

UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA
COMPUTAÇÃO

ANA LÚCIA ANACLETO REIS

**Comparação SNMP x Agentes Móveis para Gerência
de Redes**

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Ciência da Computação.

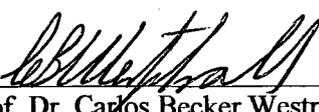
Prof. Dr. Carlos Becker Westphall

Florianópolis, novembro de 2001.

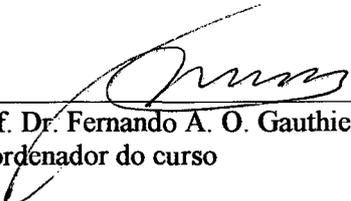
Comparação SNMP x Agentes Móveis para Gerência de Redes

Ana Lúcia Anacleto Reis

Esta dissertação foi julgada adequada para a obtenção do título de mestre em Ciência da Computação – Área de Concentração Sistemas de Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

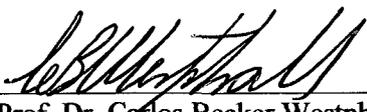


Prof. Dr. Carlos Becker Westphall
Orientador

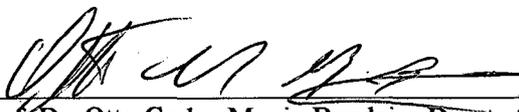


Prof. Dr. Fernando A. O. Gauthier
Coordenador do curso

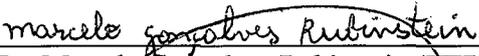
Banca Examinadora



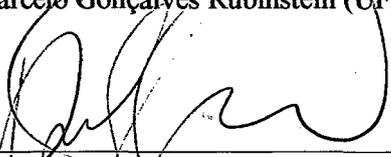
Prof. Dr. Carlos Becker Westphall



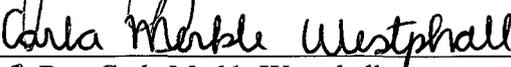
Prof. Dr. Otto Carlos Muniz Bandeira Duarte (UFRJ-GTA)



Dr. Marcelo Gonçalves Rubinstein (UFRJ-GTA)



Prof. Dr. Ricardo Felipe Custódio



Prof. Dra. Carla Merkle Westphall

“Nada na vida pode substituir a persistência: nem o talento o fará, pois o mundo está cheio de homens de talento fracassados; nem a genialidade o fará, pois gênios desprezados são quase um provérbio; nem o conhecimento o fará, pois encontraremos muitos diplomados medíocres. Só a persistência e a determinação são onipotentes”
(Calvin Coolidge)

“É muito fácil às pessoas se aproximarem das novas idéias, mas o difícil é se desfazerem das velhas”
(John Maynard Keynes)

À minha vizinha, Herondina.

AGRADECIMENTOS

Agradecimentos especiais ao Prof. Carlos Becker Westphall, pela orientação e apoio valiosos.

À Universidade Regional de Blumenau e à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pelo apoio financeiro e disponibilidade de equipamentos durante as aulas e a elaboração deste estudo.

Aos professores Sérgio Stringari, Everaldo Artur Grahl e Marcel Hugo pela confiança depositada na forma de cartas de recomendação.

Ao Prof. Ricardo Felipe Custódio e à Prof^a. Carla Merkle Westphall pelas sugestões feitas na apresentação do trabalho individual.

Ao Prof. Paolo Bellavista da Universidade de Bologna, pelos materiais tão prontamente cedidos.

Ao Hans, pela ajuda na realização dos experimentos.

Aos amigos da FURB que durante todo o tempo me apoiaram, contribuíram, incentivaram ou ainda, acompanharam de perto a elaboração deste documento: Dariva, Jomi, Marcel, Margélia, Anatoli, Marcos, Jomar, Luciano, Edson, Ana Lacene, Nilton, Fábio, Zuleika, Cibele, Elígia, Viviane, Maurizete, Bete, Izildinha, Aline, Juliana, Karen, Vicente, Michele, Homero, Tarcísio e Prof. Ivo Marcos Theis.

Às funcionárias da secretaria do CPGCC, Verinha e Valdete, pela atenção dispensada.

À toda minha família, pelo apoio dado durante todo o decorrer do curso: Mãe, Pai, Vozinha, Débora, Emerson, Juninho, Arno, Mariana, Marisa, Marcelo, Doroti, Roberto, Sybila, Raphael, Denizete, Luciano, Oswaldo, Ricardo, Fernando, Maria, Antenor, Heloisa, Vicente e Marco.

E finalmente, ao Dalton, a quem sou imensamente grata, pois sem o seu carinho, apoio e incentivo, talvez este trabalho não tivesse sido escrito ou concluído.

Sumário

LISTA DE FIGURAS.....	IX
LISTA DE QUADROS.....	XI
LISTA DE ABREVIACÕES E SIGLAS	XII
RESUMO	XIV
ABSTRACT.....	XV
1. INTRODUÇÃO.....	1
1.1. OBJETIVOS.....	4
1.2. TRABALHOS CORRELATOS.....	5
1.3. ORGANIZAÇÃO DO TRABALHO.....	7
2. GERÊNCIA SNMP	8
2.1. HISTÓRICO E EVOLUÇÃO	9
2.2. CONCEITOS DA GERÊNCIA SNMP.....	10
2.2.1. <i>Elementos ou Nós Gerenciados</i>	12
2.2.2. <i>Agentes</i>	12
2.2.3. <i>Estação de Gerenciamento</i>	12
2.2.4. <i>Base de Informações de Gerenciamento (MIB)</i>	13
2.2.5. <i>Protocolo SNMP</i>	14
2.3. OBJETOS PADRONIZADOS DE GERENCIAMENTO.....	16
2.3.1. <i>Estrutura da Informação de Gerenciamento</i>	16
2.3.2. <i>Formato das Mensagens SNMP</i>	20
2.4. ALTERAÇÕES NO MODELO VISANDO INCREMENTO DO DESEMPENHO	24
3. AGENTES MÓVEIS (AM).....	27
3.1. INFRA-ESTRUTURA	28
3.2. SERVIÇOS.....	32
3.2.1. <i>Identificação</i>	32
3.2.2. <i>Criação</i>	33
3.2.3. <i>Migração</i>	33
3.2.4. <i>Suspensão e Clonagem</i>	36
3.2.5. <i>Operação</i>	37
3.2.6. <i>Comunicação</i>	38
3.2.7. <i>Finalização</i>	40
3.3. ASPECTOS DE SEGURANÇA	40
3.4. METODOLOGIAS DE IMPLEMENTAÇÃO	42
3.4.1. <i>Sistemas de Monitoramento de Segurança</i>	42
3.4.2. <i>Sistemas de Busca de Imagens</i>	43
3.4.3. <i>Sistema de Controle de Tráfego de Veículos</i>	45

4. GERÊNCIA DE REDES COM AM	48
4.1. FORMAS DE APLICAÇÃO PROPOSTAS	49
4.2. SITUAÇÕES EM QUE O USO DE AM PODERIA APRESENTAR VANTAGENS	52
4.2.1. <i>Tarefas que geram um único resultado ou com características de autonomia</i>	<i>52</i>
4.2.2. <i>Redes com alta latência ou gargalos.....</i>	<i>53</i>
4.2.3. <i>Segmentos de Conexão Instável ou Temporária.....</i>	<i>54</i>
4.2.4. <i>Aplicações alternativas de gerência em equipamentos portáteis.....</i>	<i>54</i>
4.2.5. <i>Situações de rede desfavoráveis ou imprevistas.....</i>	<i>55</i>
4.2.6. <i>Tarefas com operações de alterações uniformes.....</i>	<i>55</i>
4.3. SITUAÇÕES EM QUE O USO DE AM PODERIA APRESENTAR DESVANTAGENS	56
4.3.1. <i>Elementos gerenciados com recursos de execução restritos.....</i>	<i>57</i>
4.4. DESENVOLVIMENTO DE APLICAÇÕES COM AM PARA GERÊNCIA DE REDES.....	57
5. AMBIENTE DE DESENVOLVIMENTO	62
5.1. SUN JAVA DEVELOPMENT KIT 1.3	63
5.2. JAVA DYNAMIC MANAGEMENT KIT 4.2.....	63
5.2.1. <i>Componentes.....</i>	<i>64</i>
5.2.2. <i>O desenvolvimento do Protótipo SNMP.....</i>	<i>66</i>
5.2.3. <i>Dificuldades encontradas</i>	<i>67</i>
5.3. IBM AGLETS WORKBENCH 1.2	68
5.3.1. <i>Componentes.....</i>	<i>69</i>
5.3.2. <i>O desenvolvimento do Protótipo de AM.....</i>	<i>71</i>
5.3.3. <i>Dificuldades encontradas</i>	<i>72</i>
6. EXPERIMENTOS REALIZADOS	74
6.1. CONDIÇÕES DE REDE TESTADAS	77
6.1.1. <i>Gerência Local</i>	<i>77</i>
6.1.2. <i>Gerência Remota – Situação 1</i>	<i>79</i>
6.1.3. <i>Gerência Remota – Situação 2</i>	<i>80</i>
6.2. ABORDAGENS DE IMPLEMENTAÇÃO.....	81
6.2.1. <i>SNMP tradicional (Protótipo A).....</i>	<i>81</i>
6.2.2. <i>AM retorna à Estação de Gerenciamento após tarefa (Protótipo B).....</i>	<i>82</i>
6.2.3. <i>AM envia dados à Estação de Gerenciamento após tarefa (Protótipo C).....</i>	<i>82</i>
6.2.4. <i>AM envia dados à Estação de Gerenciamento em um intervalo de nós visitados (Protótipo D)</i>	<i>82</i>
6.2.5. <i>Utilização de um “servidor de AM” (Protótipo E)</i>	<i>83</i>
6.3. VARIÁVEIS APURADAS	83
6.3.1. <i>Tempo de Resposta</i>	<i>83</i>
6.3.2. <i>Tráfego Total Gerado</i>	<i>84</i>
6.3.3. <i>Tráfego no Enlace de Gargalo</i>	<i>85</i>
7. RESULTADOS OBTIDOS E ANÁLISE DOS RESULTADOS	86
7.1. ANÁLISE DO TRÁFEGO TOTAL GERADO.....	86
7.2. ANÁLISE DO TRÁFEGO NA ESTAÇÃO DE GERENCIAMENTO/ENLACE DE GARGALO.....	92
7.3. ANÁLISE DO TEMPO DE RESPOSTA.....	98
7.3.1. <i>Resultados na Gerência Local.....</i>	<i>98</i>
7.3.2. <i>Resultados do 1º Experimento na Gerência Remota.....</i>	<i>101</i>

7.3.3. <i>Resultados do 2º Experimento na Gerência Remota</i>	104
7.3.4. <i>Comparações entre experimentos</i>	107
7.4. COMPARAÇÕES COM TRABALHOS ANTERIORES.....	108
8. CONCLUSÕES	110
8.1. PRINCIPAIS CONTRIBUIÇÕES	113
8.2. TRABALHOS FUTUROS	114
9. REFERÊNCIAS BIBLIOGRÁFICAS	116

Lista de Figuras

Figura 2-1: Modelo de Gerenciamento SNMP.....	11
Figura 2-2: Estrutura da informação de gerenciamento SNMP.....	18
Figura 2-3: Troca de mensagens entre os elementos gerenciados e a estação de gerenciamento.....	20
Figura 2-4: Formatos das mensagens SNMP.....	22
Figura 2-5: Esquema descentralizado de gerenciamento, proposto pelo SNMPv2.....	26
Figura 3-1: Propriedades dos agentes de software.....	28
Figura 3-2: Funcionamento tradicional de um AM (onde t = tempo).	29
Figura 3-3: Elementos de um MAS (<i>Mobile Agent System</i>).....	30
Figura 3-4: Elementos enviados no processo de transferência do AM.....	35
Figura 3-5: Arquitetura para transferência de agentes em um MAS.	35
Figura 3-6: Processo de Clonagem de um AM.....	37
Figura 3-7: Comunicação Unicast e Broadcast entre Agentes.	38
Figura 3-8: Comunicação Multicast e Forward Casting entre Agentes.....	39
Figura 3-9: Áreas de Segurança de um MAS.	41
Figura 3-10: Representação da metodologia de implementação de AM proposta por Dag Johansen.	45
Figura 3-11: Representação da metodologia de implementação de AM proposta por Schill.	46
Figura 4-1: Esquema de funcionamento de uma aplicação de gerência SNMP que utiliza AM.....	51
Figura 4-2: Esquema de implementação de AM, onde o código não é enviado de volta à origem.	58
Figura 4-3: Esquema de implementação de AM, utilizando estratégia de descarregamento.	59
Figura 4-4: Implementação de AM que utiliza servidores de agentes para otimizar operações de gerenciamento a segmentos conectados por um link mais lento.	60
Figura 5-1: Representação dos Componentes do JDMK.....	65
Figura 5-2: Interface gráfica do servidor TAHITI, componente do ambiente Aglets.....	70
Figura 6-1: Modelo de Gerenciamento de Redes utilizando AM desenvolvido por Rubinstein.....	75
Figura 6-2: Estrutura utilizada nos experimentos em rede Local.	78
Figura 6-3: Estrutura a ser utilizada nos experimentos em rede Remota – Situação 1....	80

Figura 6-4: Estrutura a ser utilizada nos experimentos em rede Remota – Situação 2....	81
Figura 7-1: Comparação entre os valores de tráfego total nos experimentos do SNMP (protótipo A) e do AM tradicional (protótipo B).....	88
Figura 7-2: Comparação entre os valores de tráfego total nos experimentos do SNMP (protótipo A) e do AM tradicional (protótipo B) em escala logarítmica.	89
Figura 7-3: Comparação entre os valores de tráfego total dos experimentos com diferentes abordagens de implementação.	90
Figura 7-4: Comparação entre os valores de tráfego total dos experimentos com diferentes abordagens de implementação em escala logarítmica.	90
Figura 7-5: Comparação entre valores de tráfego na estação de gerenciamento ou enlace de gargalo apurados nos experimentos.....	93
Figura 7-6: Comparação entre valores de tráfego na estação de gerenciamento ou enlace de gargalo apurados nos experimentos.....	95
Figura 7-7: Gráfico comparativo dos tempos de resposta obtidos no experimento da Gerência Local para o SNMP e AM tradicional.....	100
Figura 7-8: Gráfico comparativo dos tempos de resposta obtidos no experimento da Gerência Local com diferentes abordagens de implementação.....	101
Figura 7-9: Gráfico comparativo dos tempos de resposta obtidos no primeiro experimento da Gerência Remota para o SNMP e AM tradicional.....	103
Figura 7-10: Gráfico comparativo dos tempos de resposta obtidos no primeiro experimento com Gerência Remota com diferentes abordagens de implementação.....	104
Figura 7-11: Gráfico comparativo dos tempos de resposta obtidos no segundo experimento da Gerência Remota para o SNMP e AM tradicional.....	106
Figura 7-12: Gráfico comparativo dos tempos de resposta obtidos no segundo experimento com Gerência Remota com diferentes abordagens de implementação.....	107
Figura 7-13: Gráfico comparativo dos tempos de resposta obtidos no dois experimentos da Gerência Remota para o SNMP e para o AM tradicional.	108

Lista de Quadros

Quadro 2-1: Descrição dos ramos da MIB-2 e suas informações mais comuns.....	19
Quadro 2-2: Descrição dos campos que constituem as mensagens SNMP.	23
Quadro 6-1: Mapeamento dos experimentos propostos.	77
Quadro 6-2: Especificação técnica dos equipamentos utilizados nos experimentos da gerência local.	78
Quadro 6-3: Especificação técnica dos equipamentos utilizados nos experimentos da gerência remota.	80
Quadro 7-1: Valores em bytes de tráfego total obtidos nos experimentos.	87
Quadro 7-2: Valores em bytes de tráfego na estação de gerenciamento ou enlace de gargalo apurados nos experimentos.	92
Quadro 7-3: Relação dos elementos que definem o tráfego na estação de gerenciamento para o protótipo B e seus valores apurados durante os experimentos.	94
Quadro 7-4: Tempos de resposta obtidos no experimento com Gerência Local, em milissegundos.	99
Quadro 7-5: Tempos de resposta obtidos no primeiro experimento com Gerência Remota, em milissegundos.	101
Quadro 7-6: Tempos de resposta obtidos no segundo experimento com Gerência Remota.	105

Lista de Abreviações e Siglas

AM	- Agentes Móveis
API	- <i>Application Programming Interface</i>
ASN.1	- <i>Abstract Syntax Notation One</i>
aSNMP	- Agentes SNMP
CMIP	- <i>Common Management Information Protocol</i>
CMOT	- <i>CMIP over TCP/IP</i>
EGP	- <i>External Gateway Protocol</i>
FIPA	- <i>Foundation for Intelligent Physical Agents</i>
FTP	- <i>File Transfer Protocol</i>
HMP	- <i>Host Monitoring Protocol</i>
HTTP	- <i>HyperText Transfer Protocol</i>
ICMP	- <i>Internet Control Message Protocol</i>
IETF	- <i>Internet Engineering Task Force</i>
IP	- <i>Internet Protocol</i>
ISO	- <i>International Organization for Standardization</i>
ITU-T	- <i>Telecommunication Standardization Sector T of the International Telecommunication Union</i>
JDMK	- <i>Java Dynamic Management Kit</i>
JMX	- <i>Java Management Extensions</i>
MAP	- <i>Mobile Agents Platform</i>
MAS	- <i>Mobile Agent Systems</i>
MASIF	- <i>Mobile Agent System Interoperability Facilities</i>
MIB	- <i>Management Information Base</i>
OID	- <i>Object Identifier</i>
OMG	- <i>Object Manager Group</i>
OSI	- <i>Open Systems Interconnection</i>
PDU	- <i>Protocol Data Unit</i>
PING	- <i>Packet Internet Groper</i>
RFC	- <i>Request For Comments</i>
RMI	- <i>Remote Method Invocation</i>
SDI	- Sistema de Detecção de Intrusão

- SGMP - *Simple Gateway Monitoring Protocol*
- SMI - *Structure of Management Information*
- SNMP - *Simple Network Management Protocol*
- TCP - *Transmission Control Protocol*
- TMN - *Telecommunication Management Network*
- UDP - *User Datagram Protocol*

Resumo

O presente trabalho tem como objetivo apresentar uma comparação entre o modelo de gerência de redes SNMP e a proposta de uso de Agentes Móveis nas tarefas de gerenciamento de redes.

Quanto aos aspectos teóricos, o trabalho apresenta os principais conceitos da gerência SNMP e conceitos básicos sobre Agentes Móveis e suas formas de implementação. Posteriormente, mostra as propostas de integração destas duas técnicas, inclusive mencionando problemas, vantagens e situações em que esta integração pode apresentar resultados diferenciados com relação ao modelo atualmente mais utilizado (SNMP).

Na continuidade, apresenta os resultados de experimentos realizados em uma rede real, os quais utilizaram implementações desenvolvidas em Java e apuraram variáveis de desempenho como o tempo de resposta e o tráfego gerado.

Os experimentos mostraram que, apesar de gerar uma quantidade maior de tráfego na rede como um todo, o uso de Agentes Móveis pode evitar a sobrecarga na estação de gerenciamento. Além disso, comprovaram que o uso de abordagens de implementação diferenciadas no desenvolvimento de Agentes Móveis pode ampliar seus benefícios, com relação a evitar a sobrecarga da estação de gerenciamento ou enlace de gargalo e ainda com relação ao tempo de resposta.

Os experimentos também confirmaram que o tempo de passagem do tráfego de gerenciamento pelo gargalo é fator determinante tanto de melhores resultados dos Agentes Móveis quanto do pior desempenho do SNMP. Assim, o uso de Agentes Móveis é indicado para topologias onde exista significativa diferença de latência na conexão entre a estação de gerenciamento e os elementos a serem gerenciados.

Abstract

The present work has as objective to present a comparative study between the SNMP network management model and a proposal for the use of Mobile Agents in network management tasks.

In theoretical aspects, the work presents the main concepts of SNMP management and basic concepts on Mobile Agents and their forms of implementation. Later, the proposals of integration of these two techniques are described, also mentioning problems, advantages and situations where this integration can present more differentiated results in relation to the currently most used model (SNMP).

Afterwards, it presents the results of experiments carried out in a real network. These experiments used implementations developed in Java and pointed performance variables such as the time of reply and the generated traffic.

The experiments showed that the use of Mobile Agents may prevent the management station overload even though it, as a whole, generates a bigger amount of traffic in the network. Moreover, they proved that the use of different approaches of implementation may extend its benefits, preventing the management station or bottleneck overload and improving the time of reply.

The experiments also confirmed that the passage time of the management traffic through the bottleneck is a major factor for Mobile Agents better results or for worse performance of the SNMP. Thus, the use of Mobile Agents is indicated for topologies which present a significant difference of latency in the connection between the management station and the managed elements.

1. INTRODUÇÃO

A utilização efetiva das redes de computadores nas mais diversas áreas de negócios tem sido fator determinante para o sucesso das empresas da atualidade, automatizando diversas atividades e servindo como um eficiente meio de comunicação.

Motivado principalmente pelos benefícios proporcionados, o crescimento da utilização das redes de computadores tem fomentado avanços das tecnologias de interconectividade e diminuição dos custos. Tal panorama permitiu que até mesmo pequenas organizações pudessem agregar uma quantidade e variedade maior de equipamentos às suas redes.

À medida que a importância, o tamanho e a complexidade das redes aumentam, surge a necessidade de um conjunto de ações que visam tornar as redes cada vez mais estáveis e seguras, garantindo que todas as informações possam ser obtidas rapidamente. Desta forma, ações como metodologias de planejamento, procedimentos formais de avaliação, definição de serviços e metas de qualidade, segurança e, principalmente, procedimentos de monitoramento, prevenção e correção de falhas são considerados imprescindíveis.

A gerência de redes concentra principalmente as atividades descritas acima, regida por procedimentos e métodos de monitoramento, controle, administração, avaliação e planejamento. Porém, o gerenciamento de redes costumava ser simples e centralizado, quando a maioria das redes era homogênea. Hoje, existe um grande número de redes heterogêneas, o que faz com que os métodos empíricos usados até então necessitem de reestruturação (Glitho, 1998).

Controlar estruturas onde convivem diversas tecnologias, redes e serviços, equipamentos de fornecedores variados e, conseqüentemente, com prováveis protocolos e padrões distintos de comunicação são alguns dos desafios da gerência de redes. As estruturas atuais são caracterizadas pelo dinamismo de seus componentes e topologias, o que muitas vezes implica em especificações funcionais e necessidades de gerenciamento distintas (Spanhol, 2000) (Bieszczad, 1999).

Assim, para minimizar estes problemas, principalmente com relação à heterogeneidade, sugeriram esforços para adoção de padrões de atividade, que acabaram por gerar modelos de gerenciamento adotados por comunidades específicas (Glitho, 1998).

Ainda segundo Glitho (1998), a definição de um modelo de gerenciamento auxilia na redução da complexidade do problema pela “homogeneização” de alguns aspectos através da padronização. Intuitivamente, o aspecto primordial a se padronizar seria o protocolo de comunicação. Entretanto, o uso do mesmo protocolo não é suficiente para a troca de informações significativas. É necessária a padronização de arquiteturas de comunicação e informação, sem as quais as tarefas de gerenciamento em redes heterogêneas não teriam sucesso.

Neste contexto, o modelo **agente-gerente** foi proposto pela ISO (*Internacional Organization for Standardization*) na década de 80, como parte do padrão OSI (*Open Systems Interconnection*). Resumidamente, este modelo estipula que o **gerente** é uma aplicação que age no monitoramento, controle e processamento das informações recolhidas dos equipamentos gerenciados, solicitando ou recebendo informações diretamente dos **agentes**, os quais são módulos de software que residem nos elementos da rede, armazenando informações de gerenciamento.

Tal princípio também é utilizado em redes Internet (TCP/IP), onde o gerenciamento é baseado no protocolo SNMP (*Simple Network Management Protocol*), padrão atualmente mais difundido devido à popularização da rede mundial de computadores e abordado neste trabalho.

Apesar de possuírem diferenças significativas em relação aos modos de operação, os padrões citados baseiam-se no paradigma centralizado (cliente-servidor), no qual basicamente, a aplicação gerente é executada somente nas estações de gerenciamento (cliente), e utiliza protocolos de gerenciamento para recuperar informações coletadas pelos agentes instalados nos elementos gerenciados (servidores) (Rubinstein, 2001) (Susilo, 1998) (Baldi, 1997). Estas informações são processadas pela aplicação gerente e mostradas nos *consoles*¹ da equipe de gerenciamento.

¹ O nome *console* é usado para designar o teclado e o monitor do servidor (Starlin, 1998).

As interações entre gerentes e agentes são consideradas de *fina granularidade*² – também chamadas de microgerenciamento – e desta forma, podem gerar grande tráfego e sobrecarga na estação de gerenciamento (Baldi, 1997).

Este paradigma de gerenciamento centraliza o monitoramento, a análise e o controle dos dispositivos de rede, além de manter os agentes exclusivamente direcionados às tarefas de coleta e reportagem de informações solicitadas, sem haver por parte destes nenhum tipo de processamento da informação (Goldszmidt, 1998).

O paradigma centralizado de gerenciamento de redes tem sido colocado à prova pelo crescimento da escala e complexidade dos sistemas a serem gerenciados. As redes têm crescido rapidamente e têm envolvido um número cada vez maior de elementos complexos; além do montante de informações operacionais que devem ser monitoradas e processadas crescer drasticamente. Diante deste panorama, não é mais viável ou produtivo coletar e processar dados de maneira centralizada, pois a aplicação de gerenciamento pode ser inundada com informações redundantes ou sem importância, o que dificultaria ainda mais a tarefa do gerente da rede (Goldszmidt, 1998) (Costa, 1999).

A capacidade computacional dos elementos gerenciados também tem aumentado, o que torna possível que algumas funções de gerenciamento sejam processadas de modo distribuído, permitindo de certa forma, a escalabilidade do sistema de gerenciamento (Rubinstein, 1999).

A problemática da distribuição do gerenciamento também traz à tona questões que envolvem a melhoria do desempenho, o balanceamento de carga e a redução do tráfego de gerenciamento, além da confiabilidade de gerenciamento, mesmo na ocorrência de uma falha em um ponto de gerenciamento.

Vários métodos de melhoria do Gerenciamento de Redes com base na descentralização têm sido propostos e estudados, dentre eles, o uso de Agentes Móveis (AM) para a realização das tarefas de coleta e processamento das informações de gerenciamento de uma forma mais eficiente e autônoma (Baldi, 1997).

² Interações são consideradas de *fina granularidade* quando acontecem pelo envio e recebimento de informações por meio de múltiplas e pequenas mensagens (Baldi, 1997).

Resumidamente, AM são códigos agentes encarregados de realizar tarefas, que operam sobre condições específicas e podem movimentar-se através da rede, ou seja, podem ser enviados de uma máquina para outra, carregando e processando informações, inclusive cumulativas.

1.1. Objetivos

O objetivo principal deste trabalho é analisar e comparar os modelos **SNMP** e **Agentes Móveis** para o gerenciamento de redes. Este trabalho pretende apresentar os benefícios, problemas e dificuldades de cada um dos modelos, analisando aspectos como funcionamento, comportamento e desempenho, de forma que possa definir sob quais aspectos e condições os Agentes Móveis apresentam melhores resultados em relação ao SNMP.

Pretende-se validar tal estudo através da análise comparativa do desempenho dos modelos SNMP e Agentes Móveis, em experimentos realizados com diferentes abordagens de implementação na execução de tarefas de gerenciamento, em uma estrutura de rede real.

São objetivos específicos deste trabalho:

- a) conhecer os conceitos de AM com o objetivo de avaliar seu potencial de uso em aplicações de gerência de redes;
- b) comparar o modelo mais utilizado atualmente (SNMP) e a proposta que utiliza AM através de experimentos que irão apurar e avaliar variáveis de desempenho como o tempo de resposta e o tráfego gerado;
- c) comparar os resultados apurados nos experimentos mencionados acima com resultados de trabalhos anteriores, obtidos por meio de simulações e avaliação analítica;
- d) responder às seguintes perguntas de pesquisa: (1) em quais situações, dentro da Gerência de Redes, o uso de AM pode apresentar benefícios? (2) em quais situações seu uso não é aconselhado? (3) há benefícios quanto ao tempo de resposta e a economia de recursos da rede? (4) conclusivamente, seu uso é viável?

1.2. Trabalhos Correlatos

Existem várias correntes de pesquisa que têm como ponto principal de estudo os Agentes Móveis e suas aplicações, como Aridor (1998), que apresenta a infra-estrutura necessária e os requisitos para o desenvolvimento de aplicações que utilizem AM. Sistemas de Agentes Móveis e suas funcionalidades também são objetos de estudo para Baumann (1999), Braga (2000) e Gudwin (2000).

Particularmente, Johansen (1998) estuda as possíveis aplicações para este paradigma, mencionando suas possíveis vantagens em relação às alternativas atuais. O mesmo raciocínio segue Lange (1998) quando discorre sobre o futuro das tecnologias de AM.

Vários estudos têm sido desenvolvidos, no sentido de propor e validar o uso de Agentes Móveis especificamente em aplicações de Gerência de Redes.

Andrzej Bieszczad e Bernard Pagurek, ambos da *Carleton University* no Canadá, em seus trabalhos (Bieszczad, 1999) (Pagurek, 2000) apresentam propostas, aplicações e infra-estrutura necessária para o desenvolvimento de AM em aplicações de gerência de redes. Membros da mesma equipe de pesquisa, *Tony White* (1998) e *Gatot Susilo* (1998), membros da mesma equipe, também têm artigos publicados mostrando a continuidade destes trabalhos, investigando o potencial e a aplicabilidade de AM na gerência de redes.

Outros pesquisadores mantêm seus estudos focados nos esforços de tornar dos sistemas de gerência de redes mais descentralizados e flexíveis e desta forma estudam os AM como alternativas para alcançar este objetivo, cada qual propondo formas de aplicação diferenciadas. É o caso de Baldi (1997), Bohoris (2000), Glitho (1998), Rivalta (2000), Goldszmidt (1998) e Sahai (1998).

O grupo de pesquisadores italianos chefiados por Orazio Tomarchio e Antonio Puliafito da *University of Catania* (Tomarchio, 2000) propuseram uma plataforma para o desenvolvimento de AM chamada MAP (*Mobile Agents Platform*) e mostram as vantagens do seu uso, inclusive para o gerenciamento de redes. Esforços semelhantes têm sido feitos pelo pesquisador Paolo Bellavista (2001).

Otto Wittner (2000, 2000a) e Bjarne E. Helvik, da *Norwegian University of Science and Technology*, também formam um grupo de pesquisa que estuda o uso dos Agentes Móveis, especialmente no gerenciamento de falhas.

O Centro de Pesquisas em Sistemas de Comunicação da Universidade de *Surrey* também possui um projeto chamado MIAMI (*Mobile Intelligent Agent in the Management of the Information Infrastructure*) que tem, entre outras, finalidade de examinar o impacto e as possibilidades do uso da tecnologia de agentes móveis para o gerenciamento de redes e serviços (Bohoris, 2000).

No Brasil, há também pesquisadores que procuram investigar o uso de agentes móveis para a Gerência de Redes, como Marcelo Rubinstein (1999, 2000, 2001), da Universidade Federal do Rio de Janeiro, que investiga seu uso, de modo a descobrir onde e quando estes agentes devem ser utilizados. Nestes trabalhos, o desempenho da gerência SNMP foi comparado à gerência com AM por meio de simulações.

Taís Freire da Silva Costa (1999) apresentou em sua dissertação uma avaliação analítica do uso de AM na gerência de redes. Adriano Fiorese (2000) procura validar o desempenho de AM na gerência de redes através de técnicas de medidas.

O presente trabalho apresenta um comparativo entre gerência SNMP e AM, procurando enumerar situações em que os AM poderiam ser mais indicados, através da apuração de aspectos como conceitos, funcionamento e desempenho. De forma complementar, procura validar estas conclusões através da realização de experimentos em uma estrutura de rede real.

Os experimentos foram realizados em situações de rede consideradas mais adequadas ao uso de AM, conforme demonstrou o estudo teórico preliminar. Além disso, para complementar seu diferencial em relação aos experimentos dos trabalhos anteriores, os protótipos utilizaram estratégias de desenvolvimento diferenciadas, mais direcionadas ao propósito de economia de recursos e de tempo. Complementarmente, a análise dos experimentos também aborda variáveis referentes à utilização dos recursos da rede.

1.3. Organização do Trabalho

Este trabalho está organizado da seguinte maneira: o Capítulo 2 apresenta os principais conceitos da gerência de redes SNMP, apresentando aspectos como componentes e funcionamento, com o objetivo de familiarizar e fundamentar as propostas e os experimentos apresentados nos capítulos seguintes.

O Capítulo 3 apresenta a fundamentação teórica necessária para a abordagem do uso de agentes móveis, salientando sua infra-estrutura, aspectos de segurança, forma de funcionamento e ainda, apresentando métodos de implementação aplicados em outros tipos de sistemas que podem ser também utilizados em aplicações de gerência. Vale lembrar que os Capítulos 2 e 3 não têm a pretensão de esgotar o assunto, pois apresentam somente características importantes para a compreensão do trabalho.

O Capítulo 4 apresenta a proposta de Gerência de Redes com o uso de Agentes Móveis defendida por vários pesquisadores, apresentando o uso de AM como uma alternativa produtiva e vantajosa para vários tipos de aplicações, inclusive para a Gerência de Redes.

O Capítulo 5 apresenta as ferramentas utilizadas no desenvolvimento dos protótipos utilizados nos experimentos realizados com agentes móveis e agentes SNMP, que serão utilizados para a validação do trabalho.

O Capítulo 6 apresenta a metodologia de experimentos realizados para a comparação: as variáveis apuradas, as metodologias de implementação utilizadas no desenvolvimento dos protótipos e as condições de rede em que foram desenvolvidos os experimentos.

O Capítulo 7 apresenta os resultados apurados nos experimentos e a análise destes resultados. As conclusões deste trabalho, juntamente com um resumo das contribuições e sugestões para pesquisas futuras encontram-se no Capítulo 8.

2. GERÊNCIA SNMP

Existe uma grande necessidade de prover monitoramento e controle sobre todos os componentes da rede, de forma a garantir que estes estejam sempre em funcionamento e que os problemas sejam identificados, isolados e solucionados o mais rápido possível.

Para atender a esta necessidade, foram desenvolvidos modelos de gerenciamento baseados em protocolos específicos, que têm a função de fornecer mecanismos de realização das tarefas comuns, estas divididas em dois ramos, de acordo com sua natureza:

a) monitoração: sua principal ocupação é obter dados sobre o desempenho e o tráfego da rede em tempo real e, a partir destas informações, diagnosticar possíveis problemas de comunicação através de sua análise constante;

b) controle: a partir das informações coletadas, se algum problema for detectado, o gerenciamento prevê procedimentos, que podem ser preventivos ou reparadores, todos com o objetivo de sanar ou prevenir o problema da forma mais eficiente possível.

Para se tornarem mais eficientes, os modelos de gerenciamento precisam utilizar protocolos integrados, que além de facilitarem a manutenção e monitoramento, contribuem para evitar custos que seriam criados ao se utilizar soluções específicas.

Com a atenção voltada a estas necessidades e requisitos, existem várias instituições mundiais que têm se preocupado em desenvolver e aperfeiçoar estes modelos, além dos serviços, protocolos e arquiteturas que os compõem. Segundo Helden (1999), os esforços mais contundentes neste sentido foram realizados por:

a) ISO (*International Organization for Standardization*): desenvolveu o modelo de gerenciamento mais completo, seguindo o padrão OSI (*Open System Interconnection*), baseado no protocolo CMIP (*Common Management Information Protocol*);

b) ITU-T (*Telecommunication Standardization Sector T of the International Telecommunication Union*): descreveu o padrão TMN (*Telecommunication Management Network*) para gerenciamento de equipamentos de telecomunicações, sendo a aplicação do modelo da ISO em telecomunicações;

c) IETF (*Internet Engineering Task Force*): desenvolveu o protocolo de gerenciamento padrão da Internet, o SNMP (*Simple Network Management Protocol*).

O termo SNMP é atualmente usado para designar a coleção de especificações para gerenciamento de redes que incluem o protocolo propriamente dito, as estruturas de definições de dados e os conceitos associados (Stallings, 1999).

Padrão operacional “*de facto*”, a gerência de redes baseada no protocolo SNMP caracteriza-se por sua simplicidade de operação e, apesar disto, aplica-se ao gerenciamento de redes heterogêneas com as mais variadas configurações.

A simplicidade do SNMP está na definição de uma base limitada de informações de gerenciamento, com algumas variáveis dispostas em tabelas bidimensionais e um protocolo com funcionalidades limitadas, que consome poucos recursos de rede e de processamento.

Esta simplicidade permite o uso do SNMP em equipamentos bastante simples e garante a interoperabilidade entre estações de gerenciamento e softwares de fornecedores diferentes, justamente por apresentar uma estrutura bastante dirigida e restrita.

2.1. Histórico e Evolução

Inicialmente, a única ferramenta efetivamente utilizada no gerenciamento de redes foi o protocolo ICMP (*Internet Control Message Protocol*), que fornecia mecanismos para transmissão de mensagens de controle de roteadores e outros servidores para o aplicativo servidor, com a finalidade de fornecer retorno sobre problemas nos equipamentos, uma vez que servia como mecanismo de teste de comunicação entre entidades (Stallings, 1999).

Em um segundo momento, algumas ferramentas foram desenvolvidas, como por exemplo, o PING (*Packet Internet Groper*), que pode ser usado para observar variações de tempo de envio e perda de pacotes, no sentido de isolar áreas de congestionamento ou pontos de falhas.

Posteriormente, com o aumento das redes, tais ferramentas não mais atendiam as necessidades de gerenciamento. Surgiram então, ferramentas específicas para o gerenciamento, particularmente o SGMP (*Simple Gateway Monitoring Protocol*), aplicado estritamente aos *gateways*, e ainda, ferramentas mais genéricas, como o CMOT (*CMIP over TCP/IP*), o HMP (*Host Monitoring Protocol*) e o SNMP (Stallings, 1999).

Desde a sua definição inicial feita na década de 80, o SNMP evoluiu, transformando-se no protocolo de gerenciamento mais utilizado para redes baseadas em TCP/IP (*Transmission Control Protocol/Internet Protocol*), com larga aceitação que se deve basicamente à sua abordagem pragmática com relação à implementação e à definição de padrões. Enquanto outros protocolos de gerenciamento avançam lentamente através de deliberações de uma série de comitês, o SNMP já foi implementado por um grande número de fornecedores.

Com o crescimento de sua utilização, algumas deficiências do SNMP se tornaram mais evidentes, o que ocasionou o aprimoramento do modelo inicialmente definido, criando então uma segunda versão (SNMPv2), concluída em 1993 e revisada em 1995. Tal aprimoramento procurou principalmente suprir deficiências de segurança, além de adicionar um conjunto de primitivas das operações, alterações na estrutura da relação entre os gerentes e agentes e na estrutura da informação de gerenciamento (Oda, 1994) (Tanenbaum, 1997).

Finalmente, o SNMPv3 foi concluído em 1998. Esta melhoria apresenta um delineamento total para futuras versões, adicionando características de segurança ao SNMP (Stallings, 1999). Todas as melhorias mencionadas solidificaram a posição do SNMP como uma ferramenta importante no gerenciamento de redes, uma vez que todas as versões mencionadas encontram-se atualmente em uso.

Paralelamente aos esforços de melhoria do modelo, foram desenvolvidas outras técnicas que visaram descentralizar as funcionalidades de gerenciamento, como o RMON³ (*Remote Monitoring*), mas que ainda não provêem o grau de descentralização necessário à gerência de redes (Rubinstein, 2001).

2.2. Conceitos da Gerência SNMP

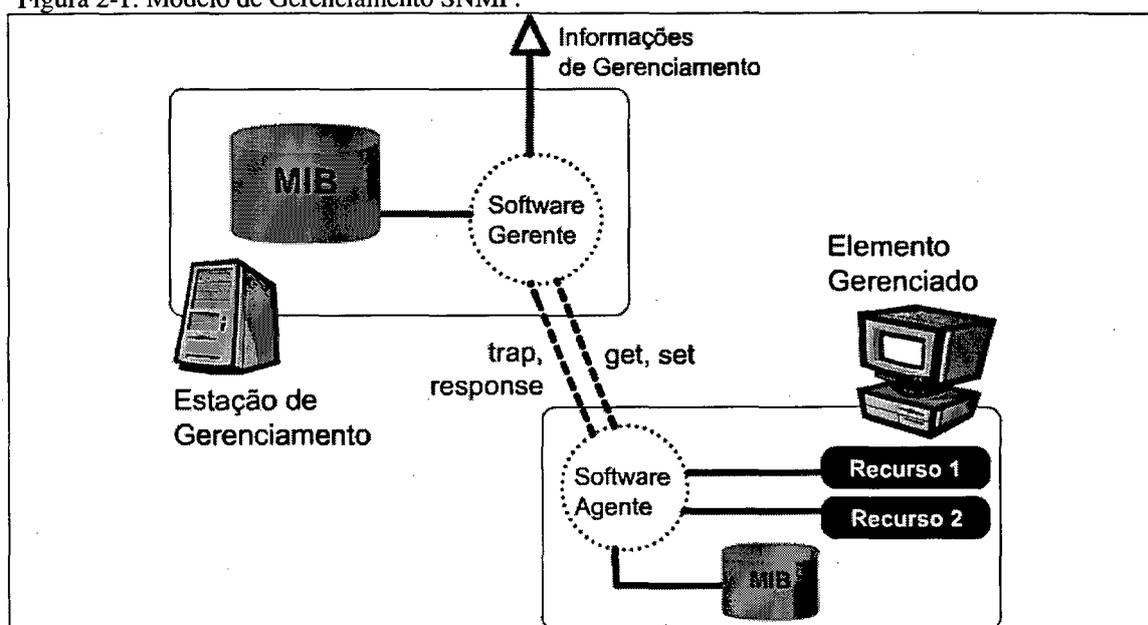
O modelo de gerência SNMP engloba o protocolo de transmissão de informações entre um ou mais sistemas de gerenciamento e seus agentes e uma estrutura para especificação e armazenamento de informações de gerenciamento (Stallings, 1998).

³ “O RMON utiliza equipamentos para realizar o monitoramento pró-ativo da rede local em segmentos locais ou remotos, de onde são providas informações sobre enlaces, conexões entre estações, modelos de tráfego e estado dos nós da rede. Estes dispositivos detectam falhas e comportamentos errôneos e geram alarmes quando identificam anomalias, mesmo não estando em contato direto com a estação de gerenciamento” (Rubinstein, 2001).

Operando na camada de aplicação, este modelo tem raízes no modelo OSI, onde os processos que implementam as funções de gerenciamento atuam como agentes ou gerentes (Martin, 1994) (Soares, 1995).

Vários autores dissertam que o modelo SNMP possui essencialmente quatro componentes: entidades ou nós gerenciados, estação de gerenciamento, informações de gerenciamento e protocolo. A gerência SNMP é realizada através de dois processos conhecidos como gerentes e agentes⁴ que interagem com estes componentes citados acima. A relação entre todos estes elementos pode ser vista na Figura 2-1.

Figura 2-1: Modelo de Gerenciamento SNMP.



Fonte: (Soares, 1995).

Conforme mostrado, o software gerente mantém as informações de gerenciamento de todo o sistema, enquanto solicita ou altera informações através de mensagens com as entidades gerenciadas. Nestas entidades gerenciadas são implementados softwares agentes, que por sua vez também mantém informações sobre cada entidade em que está instalado. Serão vistos agora todos estes elementos mais detalhadamente.

⁴ Agentes, dentro do conceito de Gerência de Redes são softwares executados em cada elemento da rede a ser gerenciado, onde sua execução é completamente controlada pelo Gerente.

2.2.1. Elementos ou Nós Gerenciados

Os elementos gerenciados representam recursos, que podem ser computadores, *hosts*, roteadores, pontes (*bridges*), impressoras, *gateways* ou qualquer outro dispositivo capaz de comunicar informações de estado para o mundo externo.

Para o modelo de gerência SNMP, cada nó gerenciado é visto como um conjunto de variáveis, que uma vez lidas e interpretadas, possibilitam a monitoração do recurso. Assim, a partir do conhecimento do seu estado, a alteração dos valores destas variáveis resume a tarefa de controle (Rose, 1994).

2.2.2. Agentes

Os agentes coletam informações relevantes junto aos objetos gerenciados e armazenam estas informações em um banco de dados local na forma de variáveis que têm a função de descrever o estado e o histórico do equipamento. Eventualmente, os agentes podem também afetar a operação dos objetos gerenciados. Para proporcionar o gerenciamento, obrigatoriamente os objetos gerenciados devem estar equipados com agentes SNMP.

Os agentes não possuem iniciativa nas ações corretivas e preventivas no processo de gerenciamento, apenas respondem às requisições de informação e às ações propostas pela aplicação Gerente, entidade que será vista na próxima seção (2.2.3). Os agentes podem ainda prover informações não solicitadas, onde o processo de reportá-las depende de regras explícitas.

2.2.3. Estação de Gerenciamento

O gerenciamento efetivo da rede é realizado na estação de gerenciamento, que é um computador que executa uma aplicação denominada **gerente**. A aplicação gerente possui um ou vários processos que solicitam e recebem dos agentes informações sobre os objetos gerenciados, que podem invocar ações ou comandos.

É no processo gerente que todo o efetivo processamento acontece, assim, este elemento é o responsável pelo processamento das informações, pela decisão de emissão de comandos e pela interpretação dos resultados (Baldi, 1997). Por este motivo, os agentes

podem ser simplificados de modo que apenas executem tarefas que lhes foram designadas, o que permite que sejam implementados facilmente nos mais diversos dispositivos.

Não existem padrões especificados quanto ao número de estações de gerenciamento existentes em um ambiente ou ainda algum tipo de média entre estações de gerenciamento e agentes. Torna-se prudente, porém, prover redundância em caso de falhas, tendo no mínimo, duas estações capazes de realizar todas as funções de gerenciamento (Stallings, 1998).

Caracterizada por sua independência de ações, a estação de gerenciamento serve como interface com o gerente humano. Seus principais componentes são: o conjunto de aplicações de gerenciamento, a interface de controle e monitoramento, a capacidade de tradução de requisitos e a base de informações extraídas de todas as entidades gerenciadas da rede (Stallings, 1999).

Especificamente analisando-se esta base de informações, considera-se que em uma mesma rede podem existir objetos gerenciados fabricados pelos mais diversos fornecedores. Assim, as informações armazenadas para cada elemento precisam seguir uma especificação rígida, de modo a facilitar a comunicação entre eles e certificar que cada elemento da rede forneça as informações necessárias ao gerenciamento de uma forma padronizada.

2.2.4. Base de Informações de Gerenciamento (MIB)

As informações sobre os recursos gerenciados são armazenadas em uma base de informações, onde estes recursos são representados por **objetos**, que por sua vez possuem aspectos inerentes aos recursos que representam.

Esta base padronizada de informações é nomeada no modelo de gerenciamento SNMP como MIB (*Management Information Base*).

Para o modelo de gerenciamento SNMP, a base de informações é padronizada, onde cada tipo de dispositivo tem um conjunto de informações que devem ser observadas, e ainda, com que formato devem ser mantidas. O formato adotado pelo modelo SNMP para codificação destes objetos é chamado de ASN.1 (*Abstract Syntax Notation One*).

Assim, a estação de gerenciamento executa sua função de monitoramento pela recuperação dos valores dos objetos da MIB, que podem gerar uma ação desta sobre um agente ou uma variável específica.

Uma descrição mais detalhada sobre a representação e o armazenamento dos dados de gerenciamento será vista na Seção 2.3.

2.2.5. Protocolo SNMP

O protocolo SNMP é utilizado para realizar a comunicação entre as estações de gerenciamento e os agentes, que estão nos elementos gerenciados.

Trabalhando em nível de aplicação, o SNMP geralmente opera sobre o protocolo UDP (*User datagram protocol*), onde originalmente nenhum controle de conexões é mantido entre a estação de gerenciamento e os agentes. Desta forma, cada troca de informações é uma transação isolada entre estes elementos (Stallings, 1999).

O processo gerente realiza o gerenciamento da rede usando SNMP que é implementado no topo dos protocolos UDP, IP (*Internet Protocol*) e outros protocolos relacionados com a parte física da rede.

Por sua vez, cada agente deve também ser implementado sob SNMP, UDP e IP. Assim, existe um processo nos agentes que deve interpretar mensagens SNMP e controlar o acesso às suas MIB's.

Basicamente, o protocolo SNMP é o responsável pelas seguintes operações de gerenciamento (Rose, 1994), (Stallings, 1999):

a) leitura das variáveis dos elementos gerenciados: realizada através do comando *GET*, onde o valor de uma determinada variável é lido e posteriormente interpretado pela estação de gerenciamento, de forma que esta possa obter detalhes do funcionamento atual do recurso gerenciado;

b) alteração de valores: os valores atuais das variáveis podem ser alterados a fim de se obter resultados – como, por exemplo, reinicialização ou tratamento de congestionamentos. Esta operação é realizada através da operação *SET*;

c) na eventualidade de acontecer situações significativas, as mesmas são reportadas pelos agentes a todas as estações de gerenciamento em uma operação conhecida como *TRAP*. Esta operação tem pretensão de simplesmente declarar que um evento ocorreu, cabendo a estação de gerenciamento procurar por maiores detalhes e executar ações cabíveis;

d) o último processo executado através do protocolo chama-se *POLLING*, que consiste em uma consulta seqüencial a cada nó, realizada em intervalos regulares, com o objetivo de detectar algum evento anormal.

Considerando especificamente esta última operação, o modelo de gerência SNMP, tipicamente centralizado, apresenta a inconveniência de não ser apropriado para gerenciar redes onde o número de objetos gerenciados é muito grande, pois o tráfego de informações de gerenciamento pode sobrecarregar a rede, causando uma situação indesejada (Mafinski, 1999).

Tal problemática permite o uso de dispositivos paliativos, como o *Trap-Directed Polling*, abordado por Stallings (1998): trata-se de um modo de funcionamento em que a operação *polling* é realizada em intervalos bem maiores, sendo em algumas ocasiões inibida.

Além de recolher as informações, o *Trap-Directed Polling* tem também por objetivo determinar uma linha-base, que passa então a ser utilizada pelos agentes, de forma que estes somente notificam o gerente se algum evento fora do comum acontecer. Uma vez notificada sobre a eventualidade, a estação de gerenciamento pode solicitar informações mais específicas sobre o caso e tomar ações corretivas.

Há casos de dispositivos que não reconhecem a pilha de protocolos necessários ao gerenciamento SNMP: são dispositivos que poderiam ter seu poder de gerenciamento limitado e que necessitam então de algum tipo de “tradutor” que façam a ponte entre as mensagens SNMP emitidas pela estação de gerenciamento e as mensagens do dispositivo, normalmente montadas em protocolos proprietários.

Para garantir que todos os dispositivos possam ser gerenciados da mesma forma, o modelo SNMP define o conceito de agente *proxy*. Este tipo de agente tem por função converter mensagens SNMP em mensagens proprietárias e vice-versa. Desta forma,

permite que as informações e ações de gerenciamento sobre o recurso sejam trocadas com a estação de gerenciamento.

2.3. Objetos Padronizados de Gerenciamento

Diante das necessidades de gerenciamento, surgiram diversas iniciativas no sentido de padronizar as estruturas de dados e o formato dos pacotes trocados entre as entidades de gerenciamento. Assim, adotou-se o documento ASN.1 (*Abstract Syntax Notation One*) que já é utilizado no modelo OSI, porém, o modelo SNMP utiliza um conjunto simplificado desta notação conhecido como SMI (*Structure of Management Information*) e ainda “algumas convenções ligeiramente alteradas” (Tanenbaum, 1997).

Estas convenções basicamente provêm a padronização das estruturas de dados e formatos das mensagens SNMP, conhecidas como PDU (*Protocol Data Unit*). As regras que descrevem as informações de gerenciamento procuraram definições que fossem independentes de implementação, como forma de garantir a padronização sob qualquer tipo de dispositivo.

Serão vistos a seguir a estrutura de armazenamento da informação de gerenciamento e o formato das mensagens trocadas entre as entidades de um sistema de gerência de redes SNMP.

2.3.1. Estrutura da Informação de Gerenciamento

Para que seja possível gerenciar os mais diversos componentes de uma rede, o modelo de gerência SNMP adota uma notação (definida pelo documento SMI), que propõe que estes recursos sejam representados como objetos. Cada objeto é essencialmente, um dado que representa um aspecto do sistema gerenciado. Como já visto anteriormente, a este conjunto de objetos é dado o nome de MIB.

Como todo o conjunto de informações é padronizado, pode-se conceber que recursos de rede de mesma função suportam os mesmos objetos. Por exemplo, todas as pontes (*bridges*) devem manter os mesmos objetos de gerenciamento, mas podem ser incluídos novos objetos para atender a necessidades de equipamentos proprietários.

Esta notação padronizada dispõe estes objetos em uma estrutura hierarquicamente organizada, onde cada elemento contém um identificador único conhecido como OID (*Object Identifier*), formado por uma seqüência de números separados por pontos e uma pequena descrição textual. Esta estrutura pode ser visualizada então como uma árvore invertida, onde cada folha adicionada é referenciada como um objeto e pode representar um componente ou uma informação armazenada sobre um recurso.

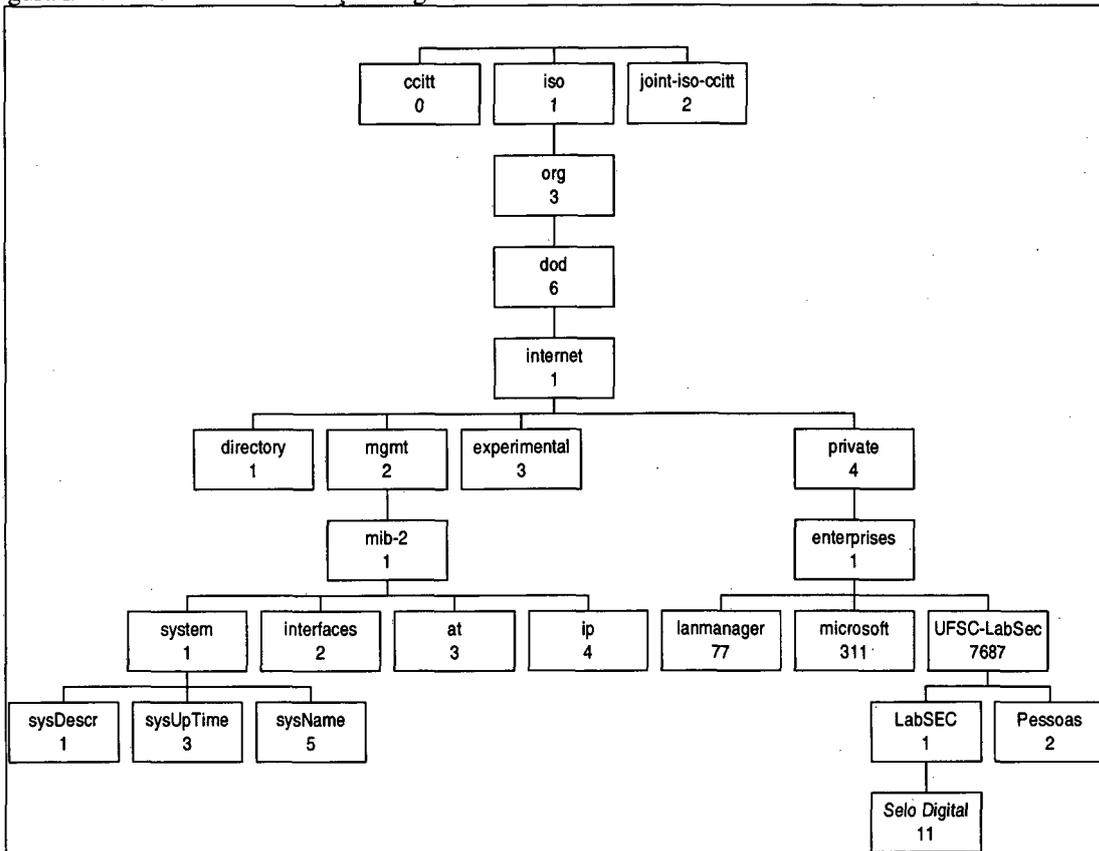
Esta estrutura é amplamente difundida, permitindo que os desenvolvedores e os fabricantes criem novos itens e atribuam um único OID para cada novo componente ou recurso, desde que autorizados pelo IETF, que é o órgão que gerencia este esquema de nomeação da estrutura da informação de gerenciamento.

A Figura 2-2 mostra a representação da estrutura conforme mencionado acima: trata-se de um subconjunto dos objetos definidos pela notação. Assim, quando um programa desejar referenciar um determinado objeto, deve informar o OID do mesmo, procedimento padronizado para qualquer sistema que siga os moldes do gerenciamento SNMP. Por exemplo, querendo referenciar a informação existente no objeto *sysDescr*, deve-se informar o OID 1.3.6.1.2.1.1.1, que corresponde exatamente à folha que contém esta informação.

Iniciando da raiz, a estrutura possui três objetos no primeiro nível: *iso*, *ccitt* e *joint-iso-ccitt*. Sob o ramo *iso*, pode-se encontrar dois sub-ramos, o primeiro utilizado por outras organizações e o segundo pertence ao Departamento de Defesa americano (*dod*). Abaixo dele, pode-se encontrar o ramo *internet*, que possui quatro sub-ramos (Stallings, 1999):

- a) *directory*: reservado para uso futuro com a OSI;
- b) *mgmt*: usado para objetos definidos para gerenciamento;
- c) *experimental*: usado para identificar objetos definidos em experimentos na internet; e
- d) *private*: mantém objetos definidos unilateralmente.

Figura 2-2: Estrutura da informação de gerenciamento SNMP.



Fonte: Adaptado de (Mafinski, 1999), (Stallings, 1999).

Abaixo do ramo *mgmt*, estão definidas até o momento, as duas versões da base de gerenciamento: *mib-1* e *mib-2*, sendo como já mencionado, a segunda MIB extensão da primeira. As duas MIB's possuem o mesmo identificador na subárvore porque apenas uma das duas estará presente em qualquer configuração. A MIB-2 (1.3.6.1.2.1) consiste em 175 objetos organizados em 10 grupos ou ramos. Alguns dos objetos mais conhecidos são listados no Quadro 2-1.

Cada objeto da MIB SNMP é declarado formalmente. Esta descrição é dividida em 5 partes: nome do objeto, sintaxe abstrata, descrição textual de seu significado, tipo de acesso permitido e o estado. O tipo de acesso pode ser: somente leitura (*read-only*), leitura e escrita (*read-write*), somente escrita (*write-only*) ou não acessível (*not-accessible*). O estado de cada objeto pode variar entre: obrigatório (*mandatory*), opcional (*opcional*) e obsoleto (*obsolete/deprecated*).

A notação ASN.1 define que os objetos podem ser declarados com tipos universais (que geralmente são mais independentes de aplicação e têm propósito geral) e ainda com

outros tipos, destinados a finalidades de aplicações especificamente e definidos pela RFC-1155 (Stallings, 1999).

Quadro 2-1: Descrição dos ramos da MIB-2 e suas informações mais comuns.

Ramo	OID	Objeto/ Descrição
System	1	Permite que o gerente obtenha informações básicas sobre o sistema local
	1.1	sysDescr Descrição do sistema: versão, hardware, sist. operacional
	1.2	sysObjectId Identificação do fabricante
	1.3	sysUpTime Tempo desde a última reinicialização
	1.4	sysContact Nome da pessoa de contato
	1.5	sysName Nome administrativo do elemento gerenciado
	1.6	sysLocation Localização física do elemento gerenciado
	1.7	sysServices Serviços oferecidos pelo dispositivo
Interfaces	2	Provêm informações sobre os adaptadores de rede e seu tráfego
	2.1	ifIndex Número da Interface
	2.2	IfDescr Descrição da Interface
	2.3	ifType Tipo da Interface
	2.4	ifMtu Tamanho do maior datagrama IP
	2.7	ifAdminisStatus Estado da Interface
	2.9	ifLastChange Hora em que a interface inicializou ou modificou seu estado
	2.10	ifInOctets Total de bytes recebidos pela interface
2.14	ifInErrors Bytes recebidos com erro pela interface	
Ip	4	Informações e estatísticas sobre pacotes IP
	4.1	ipForwarding Indica se esta entidade é um gateway IP
	4.4	ipInHdrErrors Número de datagramas recebidos descartados devido a erros em seu cabeçalho IP
	4.5	ipInAddrErrors Número de datagramas recebidos descartados devido a erros em seu endereço IP
	4.15	ipReasmOKs Número de datagramas IP remontados com sucesso
	4.21.11	ipRouteMask Máscara de sub-rede para roteamento
Tcp	6	Informações sobre o protocolo TCP (algoritmos, parâmetros e estatísticas)
	6.1	tcpRtoAlgorithm Algoritmo para determinar o tempo de expiração para retransmissão de um octeto desconhecido
	6.4	tcpMaxconn Número máximo de conexões TCP que a entidade pode sustentar
	6.10	tcpInSegs Número de segmentos recebidos inclusive com erro
	6.13.4	tcpConnRemAddress Endereço IP remoto para determinada conexão TCP
	6.14	tcpInErrors Número de segmentos descartados devido a erros de formato
	6.15	tcpOutRsts Número de reinicializações geradas
Udp	7	Estatísticas sobre o protocolo UDP
	7.1	udpInDatagrams Número de datagramas UDP recebidos
	7.2	udpNoPorts Número de datagramas UDP recebidos por falta de porta destino
	7.3	udpInErrors Número de datagramas UDP recebidos por erro, exceto devido a falta de porta destino
	7.4	udpOutDatagrams Número de datagramas UDP enviados por esta entidade

Fonte: Adaptado de (Stallings, 1999) (Mafinski, 1999) (Rose, 1994) (Tanenbaum, 1997).

São tipos de objetos universais: *Integer*, *octetstring*, *null*, *objectidentifier*, *sequence* e *sequence-of*, os dois últimos usados para construir tabelas. Também podem ser construídos tipos especiais, chamados de *enumerado*: são valores que associam um campo textual com valores numéricos e são bastante comuns. Por exemplo, um objeto deste tipo pode conter valores que são associados a estados: *up(1)*, *down(2)* ou *testing(3)*.

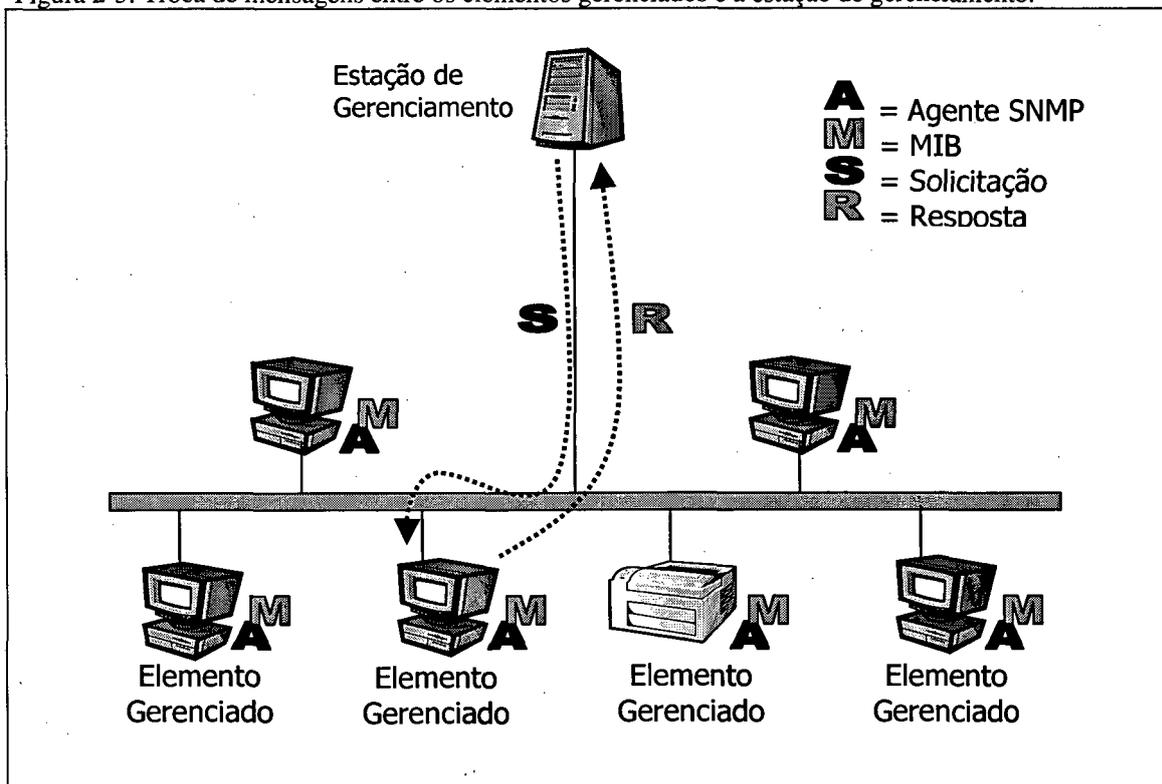
Existem ainda outros tipos que podem ser encontrados na MIB, porém, são considerados tipos para aplicações específicas: endereços IP (*ipAddress*), contadores (*counter*), medidas (*gauge*), tempo (*timeTicks*) e indefinido (*opaque*).

2.3.2. Formato das Mensagens SNMP

O funcionamento do modelo de gerência SNMP pode ser resumido (em uma visão simplista) da seguinte forma: uma estação de gerenciamento que envia solicitações a agentes contidos em nós gerenciados, fazendo perguntas sobre as 175 variáveis e sobre muitas outras variáveis específicas dos fornecedores (Tanenbaum, 1997). A Figura 2-3 ilustra esta forma de funcionamento.

O SNMP é implementado com o uso da abordagem cliente-servidor assíncrona, assim, uma entidade não necessita aguardar resposta após enviar uma mensagem, pode até mesmo, enviar outra mensagem ou continuar executando suas funções (Spanhol, 2000).

Figura 2-3: Troca de mensagens entre os elementos gerenciados e a estação de gerenciamento.



Assim, uma interação SNMP consiste em uma solicitação, seguida de uma resposta (Rose, 1994). Existem quatro formas possíveis de resultado em uma operação: uma resposta sem exceções ou erros, uma resposta com uma ou mais exceções, uma resposta com um erro ou uma expiração de tempo (*timeout*).

É importante esclarecer que não é possível através do protocolo, por exemplo, excluir ou alterar a estrutura da MIB pela inclusão ou alteração de novas instâncias de

objetos. O acesso às informações somente é realizado através da consulta às folhas da MIB. Também não é possível acessar tabelas inteiras com uma ação. Tais restrições servem para simplificar a implementação do SNMP, porém, limitam a capacidade do sistema de gerenciamento (Stallings, 1999).

Para a realização de qualquer tarefa de gerenciamento aplicada a um conjunto de elementos gerenciados, toda a troca de informações descrita acima precisará ocorrer com cada um dos elementos a serem consultados e sua estação de gerenciamento. As operações possíveis são aquelas já mencionadas (*get*, *set*, *trap*), porém, há necessidade de conhecer como esta troca de informações é realizada.

A comunicação para realização das tarefas normalmente executadas pelo modelo - quando a estação de gerenciamento envia uma mensagem a um agente solicitando informações ou forçando-o a atualizar seu estado de alguma forma; ou ainda quando o agente responde com as informações solicitadas ou confirma que atualizou seu estado da forma solicitada - obedece também a um formato padronizado.

Cinco tipos de mensagens são definidos e podem ser usados nas operações. Todas as mensagens possuem dois segmentos: um cabeçalho e uma PDU. O cabeçalho possui as seguintes informações:

- a) Versão (*version*): indica a versão do SNMP em uso, e
- b) domínio gerencial (*community*): define um dispositivo de acesso para um grupo que pode servir como uma forma de autenticação.

O último segmento da mensagem é a PDU, sua estrutura varia de acordo com a operação que se deseja proceder (Spanhol, 2000):

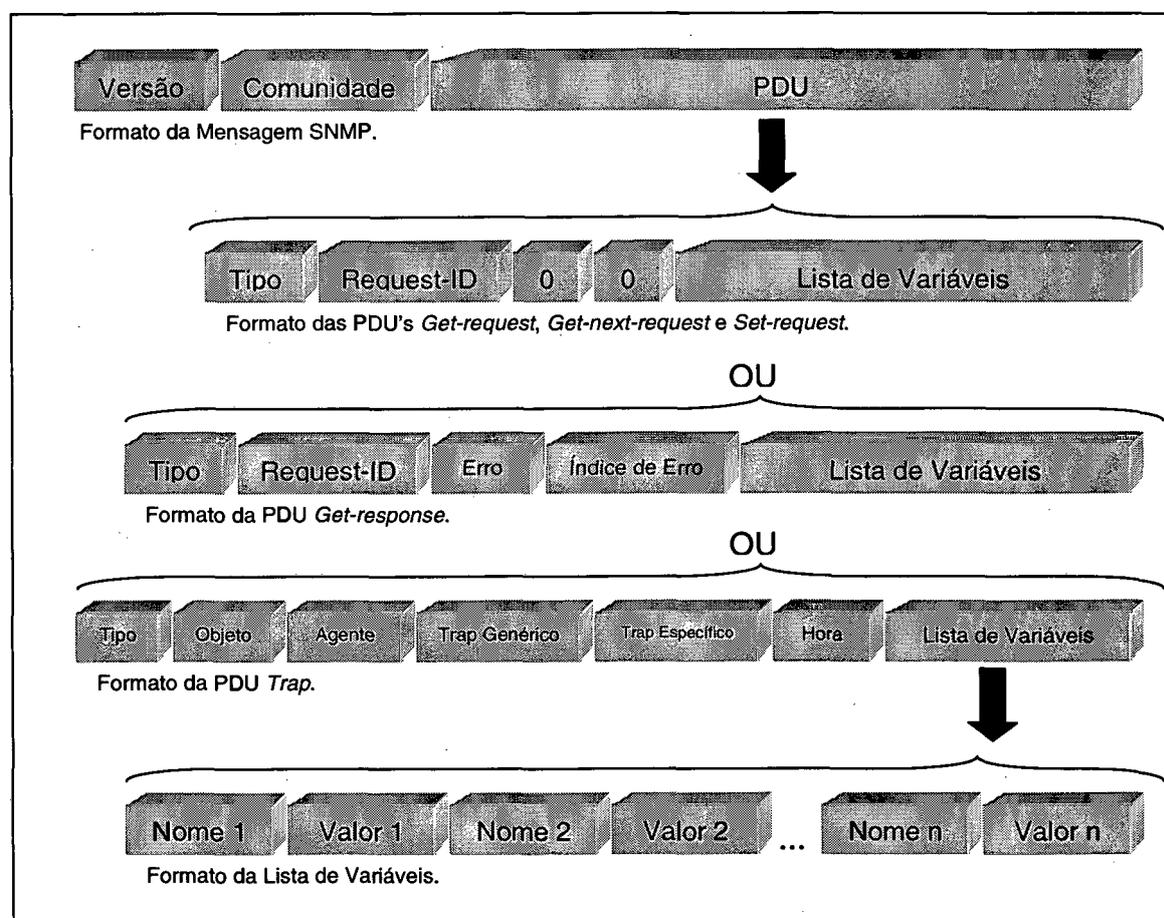
- a) *get-request-PDU*: mensagem enviada pelo gerente ao agente, solicitando o valor de uma variável;
- b) *get-next-request-PDU*: idem ao primeiro, mas solicita o valor da próxima variável, após uma ou mais variáveis terem sido especificadas;
- c) *set-request-PDU*: mensagem enviada do gerente ao agente, requisitando a atualização do valor de uma variável;

d) *get-response-PDU*: mensagem enviada pelo agente ao gerente, informando o valor requisitado de uma variável;

e) *trap-PDU*: mensagem enviada pelo agente ao gerente, notificando a ocorrência de um evento.

As representações das estruturas podem ser vistas na Figura 2-4. Os campos que constituem cada estrutura são apresentados no Quadro 2-2. É conveniente verificar que as PDU's *Get-Request*, *Get-Next-Request* e *Set-Request* possuem o mesmo formato da PDU *Get-Response*, com os valores de *erro* e *índice de erro* iguais a zero. Esta convenção reduz para um o número de formatos de PDU que uma entidade SNMP deve reconhecer e tratar (Stallings, 1998).

Figura 2-4: Formatos das mensagens SNMP.



Fonte: Adaptado de (Stallings, 1999).

Os comandos *Get-request* e *Get-next-request* são comandos emitidos pelo gerente de forma a recuperar um dado específico de um agente, em ambos os casos, se os valores

requeridos estiverem disponíveis, serão encaminhados ao gerente através de uma mensagem *Get-response*. No caso do comando *Set-request*, a confirmação de efetivação dar-se-á somente pela utilização do comando *Get-request* àquela variável logo em seguida.

Quadro 2-2: Descrição dos campos que constituem as mensagens SNMP.

Campo	Descrição	
Versão	Versão do SNMP	
Comunidade	Domínio de Gerenciamento	
Tipo	Tipo de PDU	Pode conter: 0 = <i>Get-request</i> 1 = <i>Get-next-request</i> 2 = <i>Get-response</i> 3 = <i>Set-request</i> 4 = <i>Trap</i>
Request-ID	Identificação única, especificada para cada PDU enviada	
Erro	Indica se uma exceção ocorreu durante o processamento da mensagem	Pode conter: 0(<i>noError</i>) = não houve qualquer tipo de erro 1(<i>tooBig</i>) = o tamanho da mensagem gerada excedeu o tamanho máximo permitido 2(<i>noSuchName</i>) = o valor da variável solicitada não está disponível na MIB em questão 3(<i>badValue</i>) = o valor da variável não está em acordo com a sua sintaxe, seu tipo, tamanho ou valor são inconsistentes 4(<i>readOnly</i>) = houve uma tentativa de alterar uma variável não permitida 5(<i>genErr</i>) = uma determinada variável da lista não pôde ser identificada ou alterada
Índice de erro	Quando seu valor for diferente de zero, indica qual variável da lista que causou a exceção.	
Lista de Variáveis	Lista de nomes(OID) de objetos da MIB e valores correspondentes. No caso da PDU <i>Get-request</i> , esta lista somente possuirá os nomes, sendo os valores iguais a nulo.	
Objeto	Tipo de objeto que gerou a PDU <i>trap</i> , baseado no objeto <i>sysObjectID</i> da MIB.	
Agente	Endereço ou identificação do agente que gerou a <i>trap</i> .	
<i>Trap</i> genérico	Tipo de evento	Pode conter: 0(<i>coldStart</i>) = significa que a entidade emissora do <i>trap</i> está sendo iniciada, de forma que sua configuração pode ter sido alterada 1(<i>warmStart</i>) = significa que a entidade emissora do <i>trap</i> está sendo inicializada, mas sua configuração não foi alterada. 2(<i>linkDown</i>) = a entidade emissora do <i>trap</i> reconheceu que um dos <i>links</i> de comunicação não está disponível. 3(<i>linkUp</i>) = a entidade emissora do <i>trap</i> reconheceu que um dos <i>links</i> de comunicação voltou a estar disponível. 4(<i>authentication-Failure</i>) = a entidade emissora do <i>trap</i> recebeu uma mensagem que não estava corretamente autenticada, ou seja, o nome da comunidade especificado não estava correto. 5(<i>egpNeighborLoss</i>) = a entidade emissora do <i>trap</i> avisa que a conexão com seu EGP próximo foi perdida. 6(<i>enterprise-Specific</i>) = indica que um evento específico ocorreu, normalmente a captação deste evento é pré-configurada.
<i>Trap</i> específico	Código específico do <i>trap</i> . Geralmente é usado para reportar um evento específico, configurado pelo administrador	
Tempo	Informa a hora em que o evento ocorreu.	

Fonte: Adaptado de (Stallings, 1999) (Stallings, 1998).

A partir da construção da PDU (qualquer que seja seu tipo) pela entidade SNMP transmissora, ela é passada para o serviço de autenticação, juntamente com os endereços de origem e destino e o nome da comunidade. O serviço de autenticação pode ainda realizar a

criptação⁵ ou inclusão de algum código de autenticação. Os dados resultantes vão então construir uma mensagem, sendo nesta etapa incluídas informações de versão e comunidade.

Toda a mensagem resultante é então passada para o serviço de transporte, encarregado de transferir para a entidade destino.

Já no destino, a mensagem é verificada e caso haja algum erro, é automaticamente descartada. A seguir, a versão da mensagem é comparada com a versão do destino, pois a mensagem somente será lida caso a versão seja a mesma. Os dados da mensagem são finalmente lidos e processados. A partir do processamento, esta mensagem pode gerar, por exemplo, uma mensagem de resposta.

2.4. Alterações no modelo visando incremento do desempenho

Apesar da grande aceitação pela comunidade por ser uma ferramenta relativamente simples e eficiente para o gerenciamento de redes heterogêneas, o modelo inicialmente proposto apresentou pontos fracos, apurados com relação à eficiência na transferência de grandes blocos de dados, à estratégia de gerenciamento centralizada e ainda quanto à segurança.

O modelo de gerência SNMP em sua primeira versão (SNMPv1) pode gerar um tráfego considerável durante a comunicação agente-gerente. Tudo porque nesta versão a quantidade de dados que pode ser trocada em cada transação é limitada, forçando os agentes e gerentes a gerarem múltiplas transações para realizar algumas tarefas, causando uma sobrecarga na rede que pode implicar em um aumento considerável no tempo de resposta.

Na segunda versão do SNMP (SNMPv2) foi incluído o comando *GetBulk*, que se trata de uma melhoria do comando *Get*, e tem seu foco na recuperação de valores em tabelas, que normalmente pode gerar grande tráfego. Uma tabela representa um conjunto de informações sobre um recurso ou atividade e é organizada em várias linhas. Um exemplo desta situação encontra-se no gerenciamento de um roteador, que mantém uma tabela de

⁵ Processo que torna a informação indecifrável de forma que fique protegida contra acessos indevidos, especialmente durante a transmissão ou quando os dados são armazenados. Nele, uma chave é necessária para decodificar a informação (Starlin, 1998).

roteamento onde, cada linha representa um destino diferente, sendo esta lista ordenada pelo endereço de destino.

No SNMPv1, a recuperação deste tipo de informação é feita uma linha por vez, obrigando ao gerente realizar uma série de operações *Get-request/Get-response* seguidas. Já no SNMPv2, o gerente pode recuperar toda esta informação com um único comando *Getbulk*, inclusive recuperando informações adicionais fora desta tabela com este mesmo comando.

Ainda no sentido de otimizar a transferência de dados no SNMPv2, o comando *Get* foi melhorado. Anteriormente, nos casos em que o comando *Get* precisava recuperar múltiplas variáveis e por qualquer razão uma delas não pudesse ser retornada, o comando inteiro era rejeitado e o gerente precisava reexecutar o comando com menos variáveis.

No SNMPv2, este procedimento foi reestruturado, de forma que resultados parciais possam ser aproveitados, aproveitando o valor recuperado das variáveis que não causaram o erro.

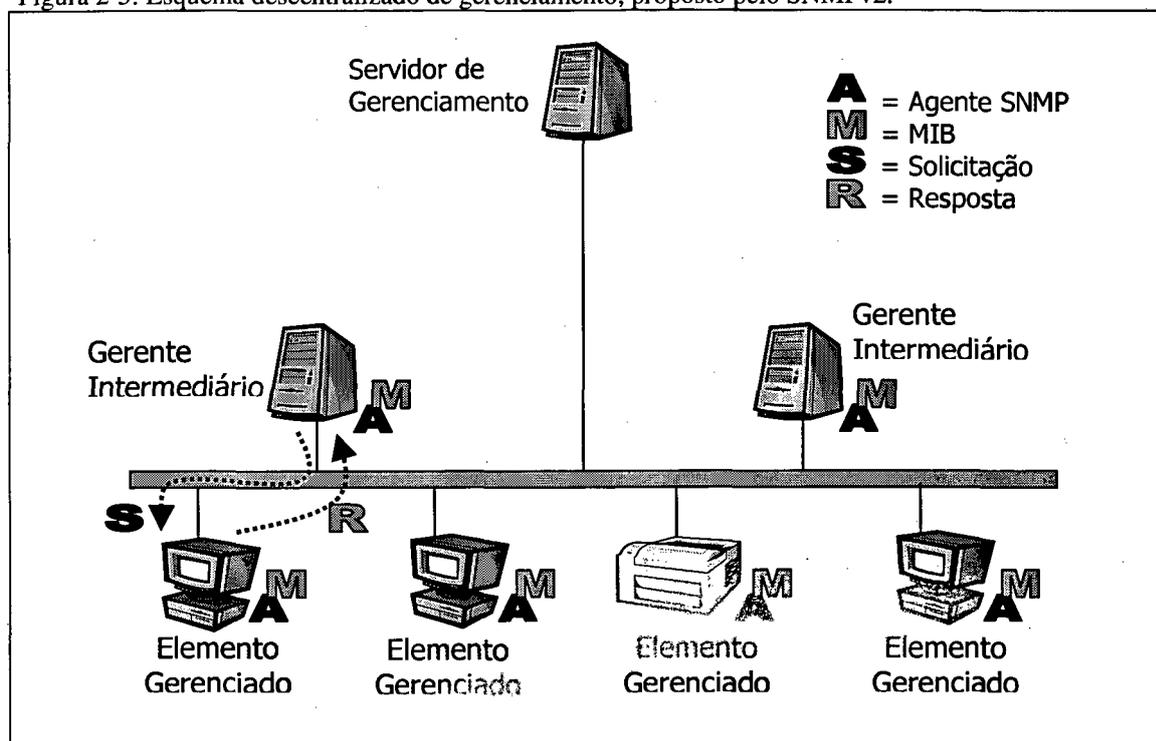
Com relação à centralização do gerenciamento, o SNMPv2 admite a existência de um gerenciamento distribuído, com estações configuradas para exercer o papel de gerentes e agentes, inclusive prevendo a possibilidade de comunicação entre gerentes para troca de informações de gerenciamento.

Esta versão prevê a existência de múltiplas estações de gerenciamento superiores, chamados de *servidores de gerenciamento*. Cada uma delas pode gerenciar diretamente uma porção do total de agentes. Existe a possibilidade de diante de um número ainda grande de agentes, o servidor de gerenciamento delegar tarefas de monitoramento e controle a gerentes intermediários.

Para que estes gerentes (servidores de gerenciamento ou gerentes intermediários) possam trocar informações, o SNMPv2 prevê dois novos conceitos: o comando *Inform-request* (um gerente remete uma informação – não necessariamente solicitada – a outro gerente) e a *MIB manager-to-manager* (base para troca de informações entre gerentes). As mensagens do comando *Inform-request* possuem o mesmo formato do comando *Get-request*.

Neste modelo, os servidores de gerenciamento podem abstrair detalhes menos importantes de um segmento de rede, sendo reportadas pelo gerente intermediário somente as mensagens mais críticas ou que geram algum tipo de necessidade de intervenção. Esta relação pode ser representada pela Figura 2-5.

Figura 2-5: Esquema descentralizado de gerenciamento, proposto pelo SNMPv2.



Tendo avaliado todos os conceitos básicos e o funcionamento da gerência SNMP, pode-se conhecer agora os conceitos de Agentes Móveis.

3. AGENTES MÓVEIS (AM)

Primeiramente, é preciso esclarecer o termo *Agente* adotado até aqui faz referência ao conceito de gerência de redes que, apesar de serem conhecidos desta forma, não possuem todas as características de *Agente de Software*, como será visto a seguir. A partir de agora, serão vistos os conceitos de *Agentes Móveis (AM)*, e os agentes descritos no modelo SNNP vistos anteriormente serão referidos como *Agentes SNMP (aSNMP)*.

Conceitualmente, um **agente de software** é um programa que age de forma autônoma em nome de uma pessoa ou organização, na maioria das vezes possuindo propriedades cognitivas, normalmente através de um procedimento interno de tomada de decisões alimentado por um conjunto de conhecimentos específicos. Resumidamente, um agente de software pode ser definido como um processo autônomo direcionado a metas (Ricarte, 2000) (Gudwin, 2000).

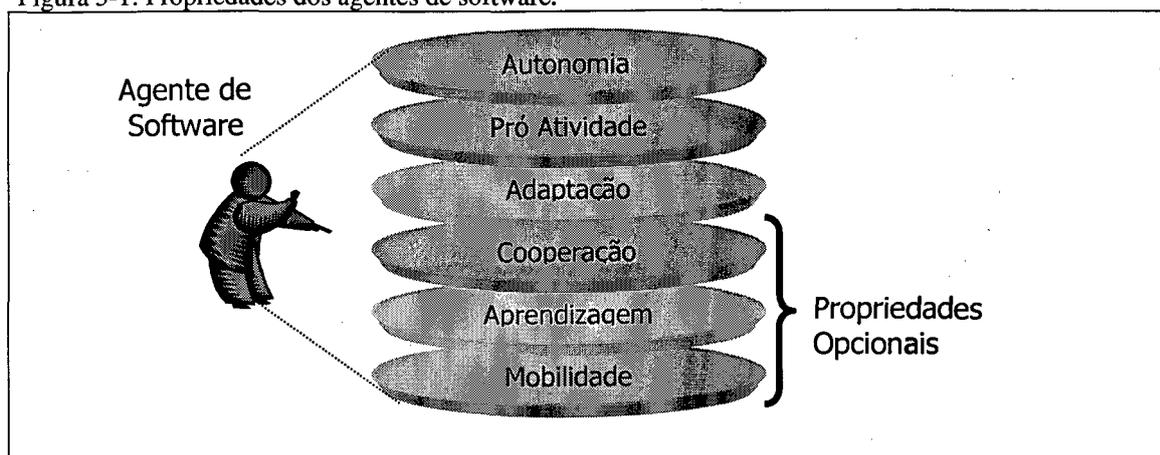
Cada agente de software é executado como um processo distinto e desta forma suas tarefas são executadas por sua própria iniciativa e sem a necessidade de intervenção contínua do usuário (*pró-atividade e autonomia*) (Milojicic, 1998).

A autonomia pode ser vista sob três aspectos: autonomia de movimento (como o AM irá se movimentar, qual seu itinerário), autonomia de execução de serviços (como sua tarefa será executada) e autonomia de aprendizado (capacidade de melhorar ou modificar seus métodos de decisão para realizar sua tarefa com maior eficiência).

Um agente de software pode reagir a situações pré-definidas e por ele detectadas (*adaptação*) e ainda, cooperar com outros agentes sempre de forma que sua tarefa seja cumprida (*cooperação*). Tecnicamente, um agente é uma instância de uma classe definida, composta de uma ou grupos de *threads*, estado e código.

Um agente de software pode ainda possuir opcionalmente a capacidade de adaptar seu processamento com base em suas experiências anteriores (*aprendizagem*) e ainda uma sexta propriedade: a *mobilidade*. Todas estas propriedades dos agentes de software são demonstradas na Figura 3-1.

Figura 3-1: Propriedades dos agentes de software.



Existem duas correntes de pesquisa de agentes. A primeira estuda agentes inteligentes e sistemas multiagentes. Seu objetivo é principalmente o estudo de agentes estacionários (fixos), distribuídos nos elementos da rede cooperando entre si para atingirem uma meta comum. A segunda corrente se preocupa com estudos de aspectos de mobilidade em agentes (Milojicic, 1999).

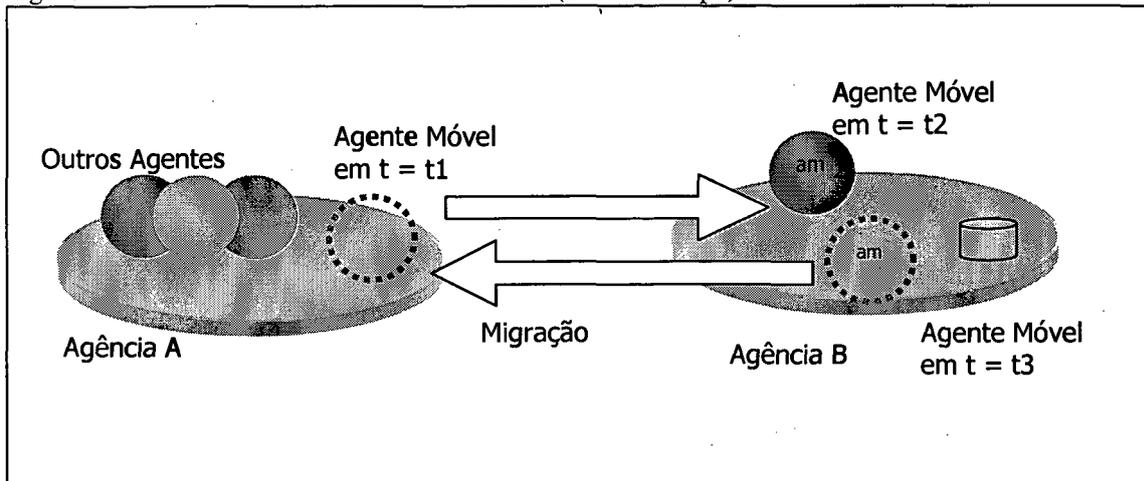
Um agente móvel (AM) é um agente de software que não se limita somente ao sistema em que iniciou sua execução: tem a habilidade de se autotransportar de um sistema para outro, juntamente com seu estado (estado de execução e variáveis) e continuar a execução de suas tarefas neste novo ambiente. Geralmente, são programados em linguagens interpretadas (como Java, TCL ou *Pearl*) devido à necessidade de portabilidade (Rubinstein, 2001).

O conceito de AM é um dos principais paradigmas da computação distribuída e vêm sendo proposto para suportar diferentes tipos de aplicações: comércio eletrônico, gerenciamento de *workflow*, gerenciamento de redes, serviços de telecomunicação e recuperação de informações distribuídas (Lange, 1998a).

3.1. Infra-Estrutura

O funcionamento típico de um AM é transferir-se para uma localização conhecida da rede, executar sua tarefa (buscar ou processar alguma informação) e depois se autotransferir para o local onde foi criado e ativado, trazendo consigo a informação processada. Este comportamento é ilustrado Figura 3-2.

Figura 3-2: Funcionamento tradicional de um AM (onde t = tempo).



Fonte: Adaptado de (Braga, 2000a).

Porém, existem outras abordagens de implementação, comentadas na Seção 3.4, que podem ser adotadas de acordo com a função do sistema a ser desenvolvido.

A **autonomia**, “aspecto herdado dos agentes estacionários, determina que as ações tomadas por AM dependem exclusivamente das informações por eles obtidas e seus padrões de raciocínio. O AM age como uma entidade independente e evolutiva, sendo capaz de coletar informações, aprender e tomar decisões sem influência externa” (Braga, 2000).

Um aspecto conceitual importante dos AM é a exigência que o código de execução tenha métodos ou recursos para invocar a sua transferência para um novo local. Ainda referindo a requisitos de implementação, o AM deve possuir a capacidade de armazenar informações e transferi-las junto com o seu código (Lange, 1998a).

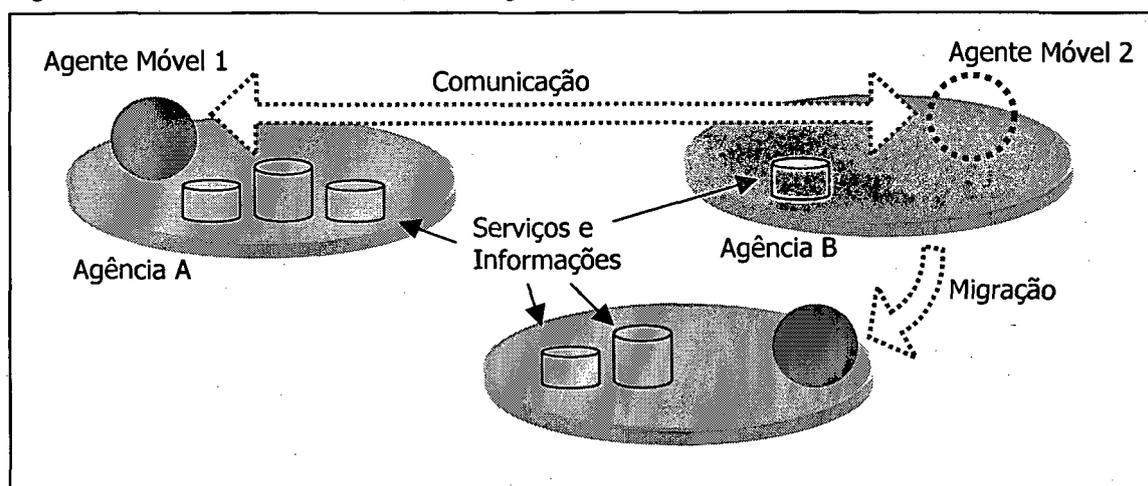
Existem outros aspectos desejáveis na implementação de AM: independência (continuar sua execução independentemente de seu agenciador) e rastreabilidade (possibilidade de ser descoberto o seu paradeiro e monitorar a sua execução) (Gudwin, 2000).

Sistemas de Agentes Móveis (MAS) ou *Mobile Agent Systems* constituem-se de uma plataforma que possibilita criar, interpretar, executar, transferir e finalizar agentes, sendo também associados a uma autoridade e identificados normalmente por um nome e endereço (Milojicic, 1999b).

Um MAS é formado basicamente por duas entidades: os **agentes** propriamente ditos e os **locais ou agências**, que são os ambientes de execução dos AM, chamados também de “*places*” (Baumann, 1998).

As **agências** têm a função de prover aos AM visitantes um ambiente seguro para execução. Geralmente, um sistema de AM consiste em várias **agências** e basicamente permitem que os AM possam mover-se entre estes ambientes disponíveis na rede, comunicando-se com outros agentes e acessando serviços específicos de cada um deles (White, 1999), como demonstra a Figura 3-3.

Figura 3-3: Elementos de um MAS (*Mobile Agent System*).



Fonte: Adaptado de (Baumann, 1998).

Um mesmo elemento gerenciado pode executar várias **agências**, que podem servir para acessar diferentes domínios de gerenciamento, com permissões de acesso a um determinado conjunto de serviços ou informações.

Dentro de um MAS, cada elemento de rede a ser utilizado possui um componente instalado que deve obrigatoriamente suprir uma arquitetura de funcionamento e alguns serviços para a execução dos AM além das agências, já citadas (Schill, 1998).

Estes serviços são funcionalidades básicas relacionadas à migração e operação do AM, como (Braga, 2000) (Aridor, 1998):

a) movimentação: funcionalidade de transferência de um AM de um sistema para outro, que consiste em uma cópia do AM para o nó destino e após a confirmação de término desta operação, o AM é destruído no nó de origem;

b) criação: funcionalidade que permite a inicialização de um AM, que pode ser invocada pelo usuário da aplicação ou ainda por outro AM;

c) clonagem: permite que um AM possa ser copiado no mesmo local ou em outro nó da rede, sendo que este novo agente é igual em dados e estado de execução ao agente clonado, porém, receberá um novo identificador;

d) finalização: uma vez que sua tarefa tenha sido cumprida, o serviço de finalização permite que o sistema possa terminar a execução do AM;

e) retração: esta funcionalidade não aparece em todas as implementações de MAS, mas permite que solicitemos o retorno de um AM que já tenha migrado para outro local. Como exemplo do uso desta funcionalidade, pode-se citar a retração de AM por razões de segurança, permitindo que estes somente migrem para um determinado equipamento caso o mesmo o solicite;

f) persistência: permite a possibilidade de desativação temporária do AM, de forma que sua execução possa ser retomada exatamente na mesma situação que o mesmo foi desativado, sem nenhum prejuízo à execução de sua tarefa ou perda de seu estado e dados já colhidos ou processados. Neste processo, o estado e os dados são armazenados em memória não volátil;

g) comunicação: funcionalidade que permite a um AM ser capaz de localizar outros AM e trocar mensagens entre si – este serviço é realizado mais comumente através de um servidor de nomes compartilhado.

Com relação à arquitetura necessária, um MAS deve possuir componentes de software que sirvam de plataforma de operação para os AM – as já citadas agências –, permitindo que os AM sejam despachados ou recebidos, além de interpretar os comandos e as funções programadas.

Os MAS podem ser implementados de forma que os AM trabalhem isolados ou em conjunto para atingir seus objetivos. Com relação à forma isolada, os métodos de implementação dos agentes devem ser mais complexos, vistos que ao trabalharem sozinhos, precisam de um grau maior de adaptabilidade e técnicas de manipulação mais otimizadas para tratar as informações colhidas, que tendem a ser maiores.

Por outro lado, em sistemas que utilizam múltiplos agentes distribuídos em vários locais diferentes e onde as tarefas são executadas em paralelo, a implementação é mais simples. Porém, surge a necessidade de implementação de mecanismos de coordenação entre agentes e prevenção da redundância nas informações obtidas.

Nestes sistemas, os AM podem trabalhar em modo não-cooperado (não precisam trocar informações entre si para realizarem suas tarefas), compartilhado (trocam freqüentemente informações entre si) ou controlado (agentes controladores que verificam as tarefas realizadas por outros agentes, como mestres e escravos) (Braga, 2000a).

Os ambientes que permitem o desenvolvimento de MAS apresentam-se como programas montados em linguagens de programação já existentes (como o JAVA, C, C++, Tcl, PROLOG, Pascal ou XML) que implementam as funcionalidades básicas para os agentes; ou ainda, como linguagens próprias com interpretadores implementados em seus sistemas (Braga, 2000a) (Costa, 1999) (Milojicic, 1999).

Há algum tempo, têm-se realizado esforços para definição de padrões para MAS. Com esta finalidade, surgiu o MASIF (*Mobile Agent System Interoperability Facilities*), que se constitui de “um conjunto de definições e interfaces que garante uma plataforma operável para sistemas baseados em agentes” (Milojicic, 1998) (Rubinstein, 2001).

Porém, mesmo diante deste padrão desenvolvido por um grupo de várias empresas e submetido à OMG (*Object Manager Group*), não existe um consenso quanto a nomenclatura utilizada, mas sim, várias funções e interfaces que definem funcionalidades básicas de um sistema de agentes móveis (Braga, 2000) (Gudwin, 2000).

3.2. Serviços

Serão vistas agora mais detalhadamente cada uma das características dos serviços de um MAS disponíveis aos AM.

3.2.1. Identificação

Em um sistema onde seja prevista a execução de vários agentes, é necessário identificar unicamente cada AM, de forma a permitir operações de gerenciamento e localização. Assim, os AM são normalmente identificados por seu nome de autoridade

(pessoa ou organização proprietária do agente), código identificador e tipo de sistema, compondo um **identificador único** (Milojicic, 1999b).

Para garantir a identificação sem duplicatas, em determinados ambientes, os AM podem ser identificados também com informações como: o endereço de rede onde foi criado e hora em que foi criado. Outros ambientes, são menos exigentes, e procuram identificar seus agentes apenas com um identificador de rede e um identificador de agente, o que pode acarretar em uma probabilidade maior de gerar dois AM com o mesmo identificador, algo não desejável, pois pode gerar confusões no envio de mensagens e localização destes AM.

3.2.2. Criação

O processo de criação de um AM resume-se em requisitar a uma agência ou sistema de Agentes Móveis o início da execução de um AM. Este processo pode ser solicitado por um usuário, disparado por um evento externo ou até mesmo por outro AM. Quando na ocasião de sua criação, o AM armazena informações sobre sua identidade e sua agência origem (Milojicic, 1999b).

Toda a operação de criação de um AM deve apontar à agência o local do código fonte do AM a ser criado, que pode não estar localmente armazenado. Alguns sistemas admitem a existência de criações sincronizadas em múltiplos locais com objetivo de execução de uma tarefa específica (Braga, 2000a).

Ainda na operação de criação de um AM, seu criador pode especificar o conteúdo de suas variáveis, permitindo que algumas tarefas sejam executadas por agentes previamente parametrizados.

3.2.3. Migração

A mobilidade aparece como característica fundamental dos AM sendo que estes podem iniciar sua transferência para outro local pela execução de uma instrução específica em meio ao seu próprio código. Paralelo a isso, suas informações são salvas e também transferidas ao novo destino. Já no destino, o programa é reiniciado e a execução deve conservar as mesmas informações que tinha na origem, que forma que possa continuar a execução (White, 1999).

A movimentação pode ser realizada a partir de duas ações distintas: a primeira, chamada de translação, é a transferência com base na realização de uma cópia no destino e a destruição do AM na origem; a segunda chama-se retração, e consiste na solicitação de retorno de um AM que já migrou para outro local (Braga, 2000a).

A retração não é uma característica que aparece em todos os sistemas de agentes móveis e sua principal utilização acontece em situações onde haja dispositivos de segurança que impedem que os AM retornem à sua origem, como em redes equipadas com *firewall*.

A forma como o estado e informações são transferidos (durante o processo de migração) e reativados pode assumir dois padrões (Baumann, 1999) (Braga, 2000a):

a) *migração forte ou transparente*: quando o AM reinicia a execução em seu novo destino exatamente com o mesmo estado e na mesma posição de código que ele estava executando antes de sua migração;

b) *migração fraca ou não-transparente*: o código do AM é transferido juntamente com seus dados para o destino, porém, o estado de execução não é transferido.

Em determinadas linguagens de desenvolvimento, como o Java, o AM precisa preparar sua migração, armazenando seu estado e informações necessárias em variáveis e já no novo destino, precisa também iniciar sua execução a partir da leitura destas variáveis para restaurar seu estado. Neste modelo, na verdade o estado do sistema é simulado, assim, informações específicas do processo de execução do programa (pilha, ponteiro do programa, e outros itens) não são transferidas. É importante ressaltar que existem esforços em modificar a máquina virtual Java para prover a mobilidade forte (Suri, 2000).

O processo de transferência de um AM inicia com o pedido, feito por parte do próprio AM que deseja se transferir. A partir deste pedido (que contém também o endereço do destino desejado), a agência suspende a execução do agente e envia para a agência de destino uma requisição de transferência.

Durante o processo de transferência dos agentes, o local de destino e a qualidade do serviço de comunicação são identificados. Quando o sistema destino aceita a transferência, o estado do agente, juntamente com sua autoridade, credenciais de segurança e - caso haja

necessidade - o código são transferidos para o destino, conforme ilustra a Figura 3-4. Esta transferência é realizada pela codificação destas informações em um protocolo de transferência prédefinido. Já no destino, a agência decodifica os pacotes recebidos, reconstrói o processo do agente e reinicia sua execução (Milojicic, 1999b). Todo este processo pode ser resumido pelo esquema apresentado na Figura 3-5.

Figura 3-4: Elementos enviados no processo de transferência do AM.

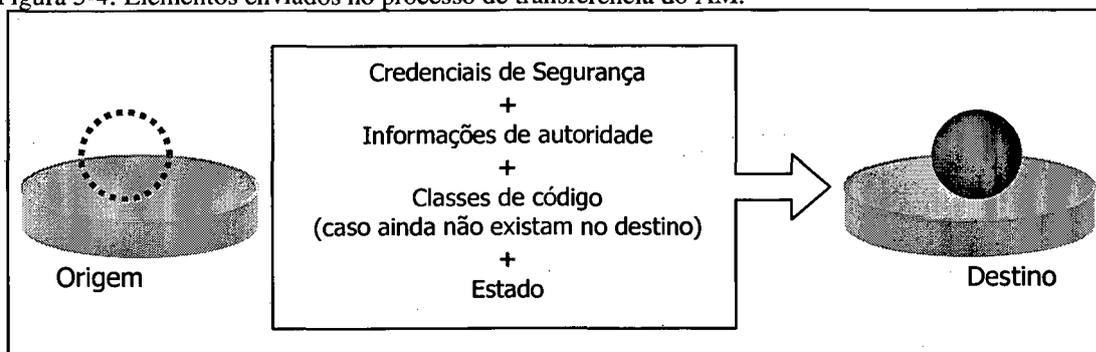
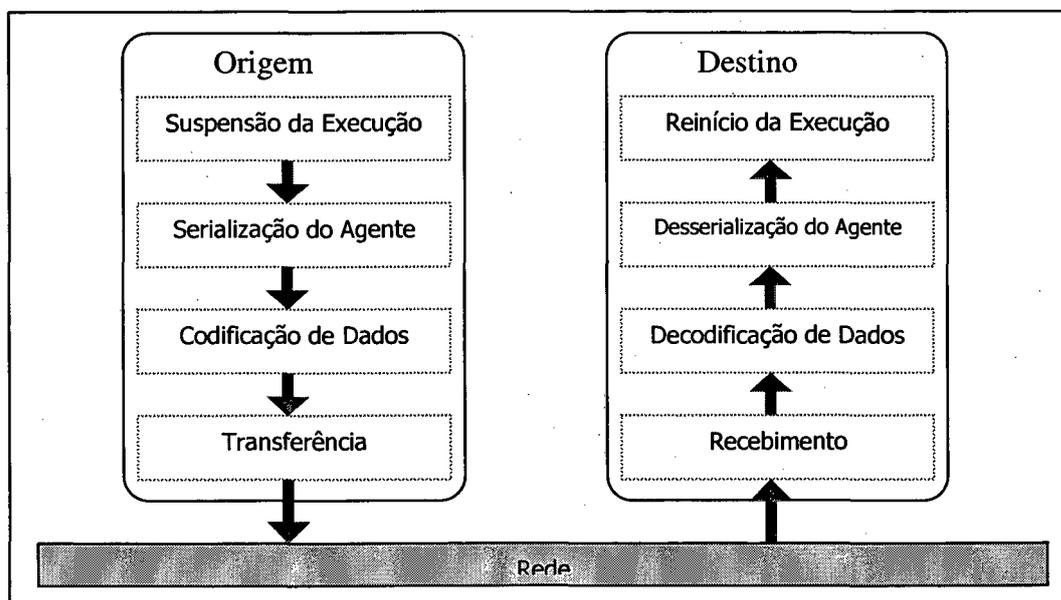


Figura 3-5: Arquitetura para transferência de agentes em um MAS.



Fonte: (Ricarte, 2000) (Lange, 1998a).

A transferência de classes – que em um MAS são normalmente fornecidas por um equipamento **provedor de classes** – pode ser realizada de três formas (Braga, 2000a) (Milojicic, 1999b):

a) **automática**: todas as classes necessárias são transferidas com o AM a cada vez que ele muda de local;

b) **sob requisição:** são enviadas as classes referentes à instância do AM e a partir daí, caso alguma outra classe seja necessária, ela será requisitada e remetida ao destino;

c) **sob informação:** na ocasião da transferência, a origem envia também uma lista com a relação de todas as classes necessárias, o destino verifica se já as possui e em caso negativo, solicita as inexistentes. Esta forma pode variar se a transferência for feita para um destino que não permanece sempre conectado ou ainda, quando acontece um pedido remoto de criação de um agente, quando então automaticamente todas as classes são enviadas, dispensando esta troca prévia de informações.

Obviamente a forma automática de envio de classes consome banda passante desnecessariamente, visto que o destino pode receber classes que já se encontram armazenadas nele.

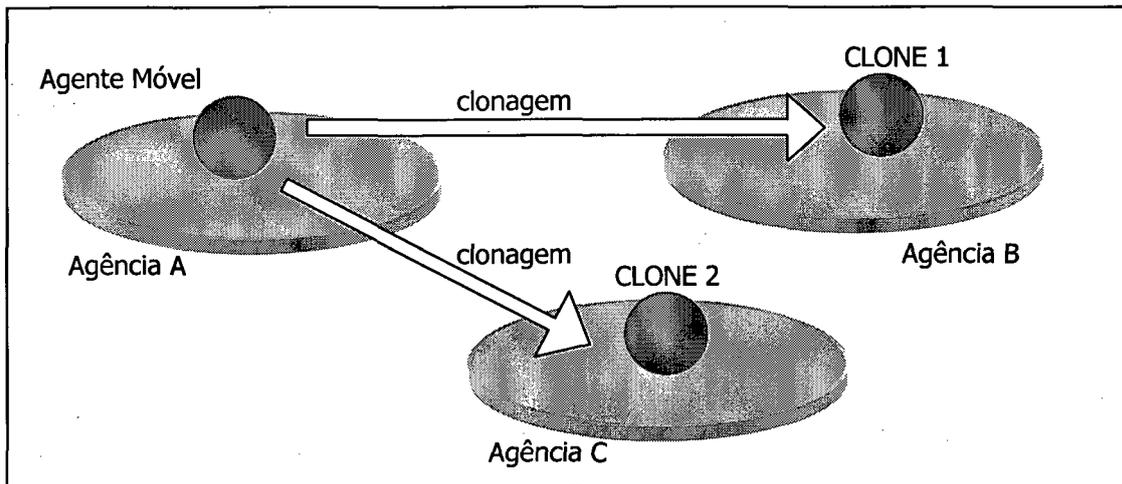
No caso do provedor de classes não estar disponível no momento da requisição, existe a possibilidade do MAS solicitá-las diretamente ao local de origem do AM recebido.

3.2.4. Suspensão e Clonagem

Durante todo o tempo de sua execução, um AM pode submeter-se a procedimentos que suspendem temporariamente sua execução, que variam de acordo com sistema utilizado. Os dados pertencentes ao AM podem ser armazenados e são recuperados quando da ocasião do reinício da execução (suspensão persistente). Porém, sistemas mais simples podem implementar funcionalidades de suspensão que não preservam os dados (suspensão volátil) (Lange, 1998a).

Durante sua execução, um AM pode ser clonado ou replicado, ou seja, é realizada a criação de uma cópia do AM, que será igual em dados e estado de execução, mas terá um novo identificador. Geralmente, este método é utilizado a partir de uma solicitação de serviço, quando o AM pode enviar uma cópia de si mesmo para realização da tarefa sempre que solicitado. A representação deste processo pode ser vista na Figura 3-6.

Figura 3-6: Processo de Clonagem de um AM.



Fonte: Adaptado de (Braga, 2000a).

3.2.5. Operação

A operação de um AM pode assumir padrões diferentes, o que pode variar de acordo com a tarefa que deve realizar, como segue (Costa, 1999) (Braga, 2000) (Lange, 1999):

a) reativa: um AM pode modificar seu padrão de operação a partir da ciência de um evento que implique em mudança de ambiente ou mesmo devido ao recebimento de uma mensagem proveniente de uma agência ou de outro AM;

b) espontânea: o comportamento espontâneo do AM é definido em seu código quando, por exemplo, em um dado momento, resolve executar uma busca ou uma determinada seqüência de ações a partir de seu próprio mecanismo de raciocínio ou detecção;

c) pró-ativa: agentes podem ser programados para executar ações que implicam em modificações em seu ambiente, que refletirão inclusive em suas próximas ações, sem a necessidade de receber uma ordem ou ter detectado alguma anomalia.

Vale ressaltar que os comportamentos citados acima podem aparecer em um mesmo agente, que pode ter sua operação direcionada com padrões diferentes que dependerão da situação que se encontra.

3.2.6. Comunicação

Como já mencionado, a comunicação entre AM é um dos serviços que devem ser providos por um MAS e serve para que os agentes troquem dados e informações de forma a realizar sua tarefa com maior eficiência, inclusive provendo ações coordenadas (White, 1999).

A comunicação entre AM é provida na maioria dos sistemas com métodos ou funções invocadas em nível de linguagem de programação. As mensagens podem ser transmitidas de quatro maneiras (Braga, 2000a): *unicast* (um AM utiliza a identificação para enviar a mensagem diretamente ao AM desejado), *broadcast* (o AM envia a mensagem para todo o grupo de AM, os quais irão interpretar e verificar se a mensagem é válida naquele contexto que se encontra), *multicast* (os AM são inscritos em grupos de serviços, que são o alvo das mensagens) e *forward casting* (a mensagem é enviada para o AM mais próximo, o qual por sua vez a repassa até que chegue ao seu destinatário). O funcionamento destes métodos é representado na Figura 3-7 e na Figura 3-8.

Figura 3-7: Comunicação Unicast e Broadcast entre Agentes.

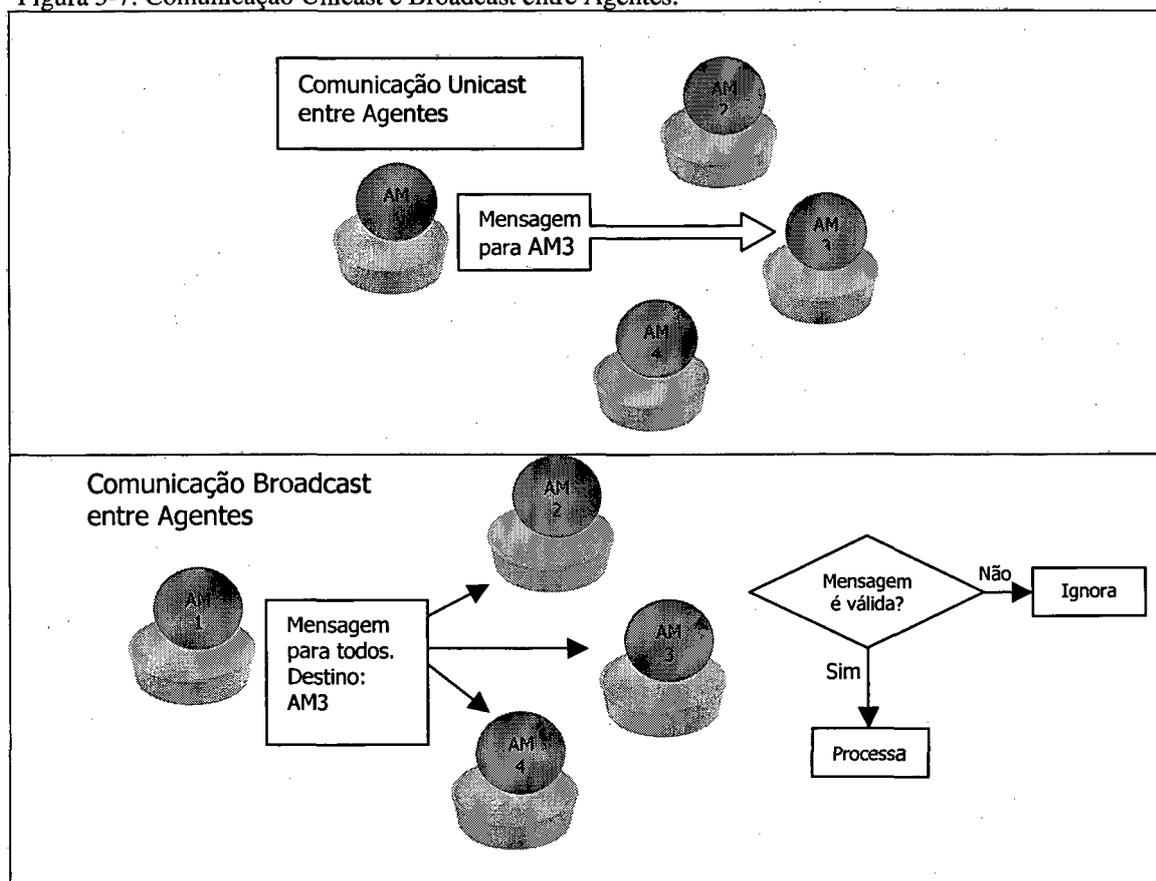
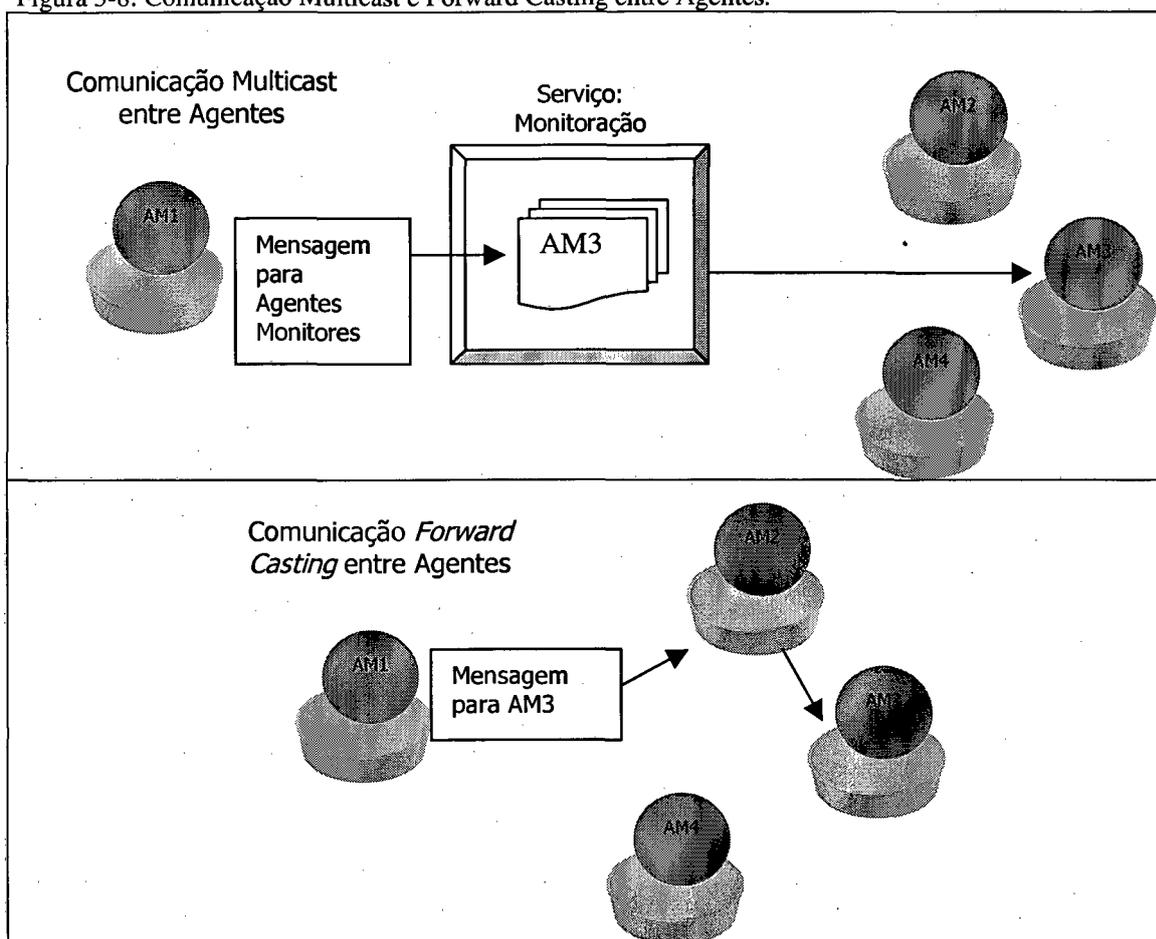


Figura 3-8: Comunicação Multicast e Forward Casting entre Agentes.



Alguns ambientes implementam a funcionalidade de comunicação de forma síncrona, ou seja, o AM obrigatoriamente deve suspender sua execução e aguardar a resposta de sua mensagem. Em contrapartida, outros ambientes tornam esta funcionalidade mais produtiva e permitem que os AM continuem executando suas tarefas enquanto não recebem sua resposta (Lange, 1998a).

A comunicação em um MAS também pode variar de acordo com a forma de endereçamento, ou seja: direta (os AM enviam mensagens diretamente uns aos outros), indireta (as mensagens são enviadas às agências, que se encarregam de repassá-las ao destinatário) ou através de mecanismos que funcionam como uma caixa postal, que são repositórios de mensagens periodicamente consultadas pelos AM (Braga, 2000a).

3.2.7. Finalização

A finalização de um AM, que é o processo onde a execução do AM é terminada e conseqüentemente seus dados são perdidos, pode acontecer pelos seguintes motivos (Lange, 1998a):

a) término da tarefa;

b) redundância (quando dois AM percebem que estão realizando a mesma tarefa no mesmo local, um deles pode decidir por sua finalização);

c) aniquilação (um agente de software regulador detecta que um AM deve ser finalizado, com base em diretivas para evitar super população ou execução com uso excessivo de recursos), e

d) iniciativa própria (quando, por exemplo, o AM detecta que seu objetivo não pode ser alcançado).

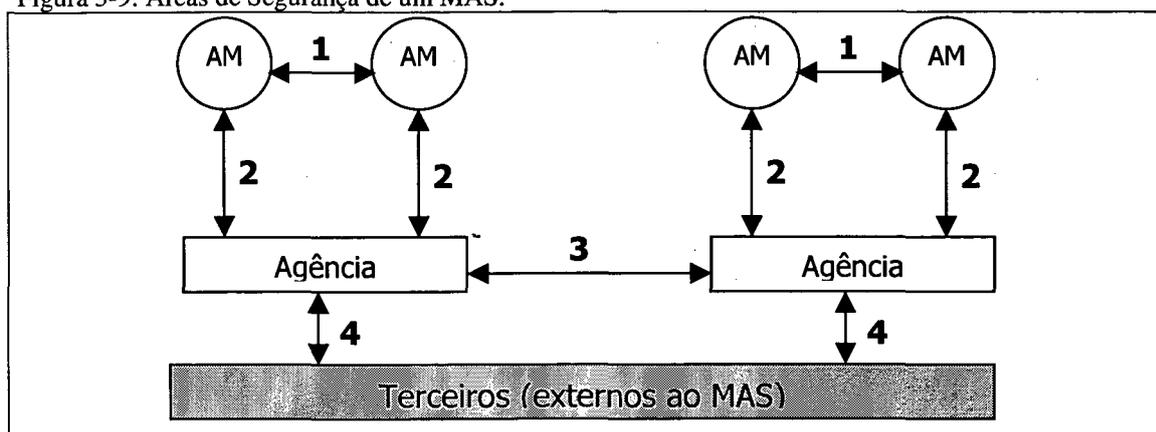
Da mesma forma como o processo de criação, a operação de finalização de um AM pode ser solicitada por um usuário, disparada por um evento externo ou até mesmo por outro AM.

3.3. Aspectos de Segurança

É crescente a preocupação com os agentes criados por invasores para ocasionarem danos às informações ou vazamento de informações sigilosas (Milagres, 2001) (Costa, 1999).

Baumann (1999) classifica a segurança de MAS em quatro áreas distintas que devem ser alvo de preocupações na ocasião do desenvolvimento, representadas na Figura 3-9: inter-agente (1), agente-agência (2), inter-agência (3) e agência-terceiro (4).

Figura 3-9: Áreas de Segurança de um MAS.



Fonte: (Baumann, 1999).

No desenvolvimento de um MAS, deve haver uma preocupação constante quanto à tentativas de violação (acesso não autorizado) ao código e ao conteúdo das variáveis carregadas pelo agente, no sentido de prevenir alterações maliciosas neste conteúdo – os conhecidos “cavalos de tróia” – trazendo problemas para a estrutura de recursos de uma rede (sistema operacional, sistema de arquivos, discos, CPU, memória, programas locais) (Milojicic, 1999b).

Assim, o MAS deve identificar e monitorar todos os agentes visitantes, verificando sua identidade e a autoridade que enviou o agente, e desta forma, fornecendo privilégios de acesso condizentes.

Para garantir que os AM não contenham instruções não autorizadas, conjuntos de regras devem ser definidos como políticas de segurança que governam todas as atividades. Tanto os AM quanto o MAS podem possuir vários conjuntos destas políticas, que contêm regras que restringem ou concedem capacidades aos agentes, podem configurar seu limite de consumo de recursos, ou ainda restringir ou conceder acesso.

Para que a utilização destas políticas seja eficiente, é necessária a implementação de um sistema de autenticação, que possibilite ao MAS armazenar e utilizar informações de autenticação de acordo com a autoridade, além de validar o código de um AM recebido, de modo a certificar-se que seu código original não foi adulterado.

As maiores vulnerabilidades de um MAS podem estar na infra-estrutura de comunicação (Milojicic, 1998). Assim sendo, métodos de autenticação também devem ser previstos para a realização destes serviços. Milojicic (1999) acredita que, apesar da

apreensão inicial, o uso de AM pode se tornar bastante seguro, desde que as devidas precauções (citadas acima) sejam tomadas.

3.4. Metodologias de Implementação

Várias são as situações em que os AM podem ser aplicados e que provavelmente, contribuiriam ou contribuem de forma significativa. Um MAS pode ser implementado de diversas formas, de acordo com a situação de rede na qual ele será implantado, o que permite que se opte por alternativas de implementação diferentes da mais clássica – onde um AM é programado com suas tarefas, despachado na rede, para depois retornar à origem com os resultados – o que pode aumentar seus benefícios.

O objetivo dos exemplos abaixo é ilustrar as soluções adotadas por pesquisadores – principalmente com relação à metodologia de implementação adotada ou sugerida – e que também podem trazer vantagens a aplicações de Gerência de Redes.

3.4.1. Sistemas de Monitoramento de Segurança

Francisco Milagres (2001) realiza pesquisas do Laboratório de Intermídia da USP (Universidade de São Paulo), nas quais se propõem soluções que aplicam AM como incremento aos sistemas de segurança computacional, propondo assim, mecanismos de monitoração e notificação munidos de mobilidade.

Uma destas pesquisas, desenvolvida por Mauro Bernardes, propõe que os AM sirvam como base para o desenvolvimento de Sistemas de Detecção de Intrusão (SDI), adicionando a estes sistemas capacidade de tolerância a falhas e flexibilidade, além de minimizar problemas conhecidos com relação à capacidade de configuração, escalabilidade e eficiência (Milagres, 2001).

Tomando o lugar do atual sistema centralizado, esta nova abordagem dá espaço a um sistema de segurança modular, baseado em agentes autônomos e móveis, que poderiam agir independentemente, observando o comportamento do sistema, trocando informações e notificações no caso de detecção de alguma atividade suspeita, e ainda, realizando ações reativas.

Como vantagem pode-se citar a relativa tolerância a falhas, particularmente falando-se da segurança contra ataques tipo *Denial of Service*, que se configuram como ataques direcionados à máquina que hospeda o SDI.

Esta abordagem sugere que “*cada agente observe somente um pequeno aspecto de todo o sistema*”, tornando este flexível à ajustes necessários nos casos de descoberta de novas formas de intrusão, pois seria necessária somente a inserção de novos agentes, em contraposição à reconstrução de parte do sistema tradicional e às paralisações temporárias que estas reconstruções conduzem. Da mesma forma, se houver algum tipo de alteração de comportamento de intrusão, somente o agente correspondente precisaria ser excluído ou alterado.

A utilização de agentes autônomos e móveis neste tipo de sistema também o torna mais simples, entretanto, a proposta apresentada prevê que seriam necessários agentes com graus de especialização diferentes para cada tipo de ataque, que seriam acionados quando qualquer agente do sistema localizasse algo que indique uma intrusão. Existiriam, nesta abordagem, agentes de vigilância (para coleta de informações), de tomada de decisão (para análise e identificação de possíveis intrusões), de notificação (para notificar o administrador da rede) e de reação (para execução de ações de contra-ataque), todos cooperando entre si.

A mesma idéia é utilizada no trabalho de White (1998), que propõe estrutura semelhante para a localização de falhas em redes de computadores com agentes móveis inteligentes equipados com mecanismos de detecção de falhas e outros com mecanismos de correção ou prevenção de problemas, atuando de forma cooperada, constituindo também um modelo de sistema promissor.

3.4.2. Sistemas de Busca de Imagens

Uma outra contribuição dos AM é apresentada por Dag Johansen (1998), que propõe um sistema “servidor de imagens” para ser utilizado na otimização da recuperação de informações de satélites, diminuindo consideravelmente o tráfego de rede.

Normalmente, satélites capturam imagens cujo tamanho varia em *terabytes*, e por isso, são extremamente problemáticas de serem transmitidas. Os AM melhoraram a troca

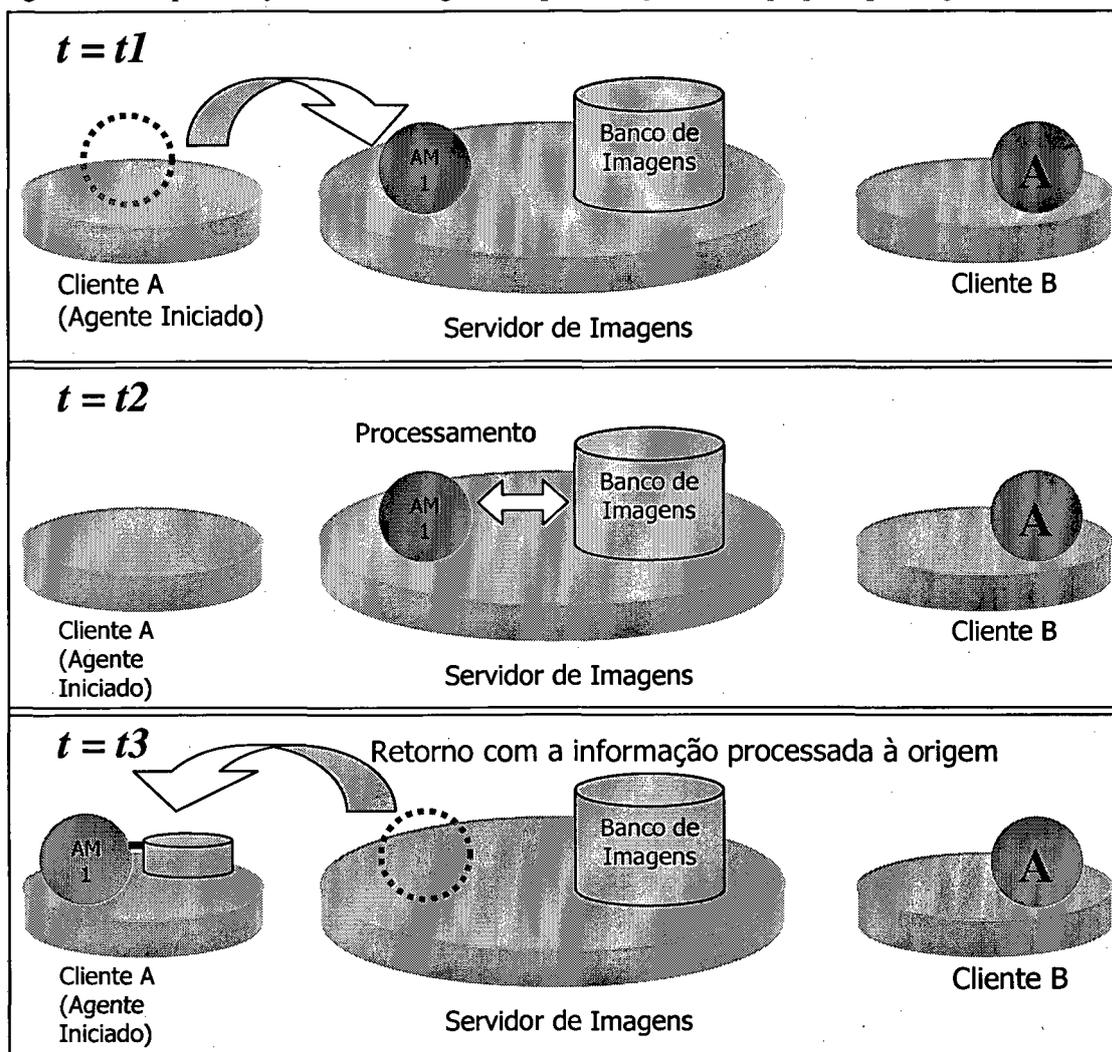
destas informações; através de um modelo onde os clientes que necessitam de informações, precisam iniciar um AM que é transferido até o servidor de imagens.

A partir daí, todo o processamento das instruções contidas no agente seria realizado no servidor e somente as informações solicitadas seriam enviadas ao cliente. Como por exemplo: a filtragem de informações de uma área geográfica específica, ou ainda, a contabilização a partir de uma imagem, da quantidade de *pixels* de uma certa cor, que podem representar neve ou certas condições de tempo. Esta metodologia é representada pela Figura 3-10.

O trabalho citado também apresentou experimentos com o modelo desenvolvido, que resultaram em consideráveis melhorias na diminuição do tráfego de rede gerado e também no tempo de resposta, que ficaram muito abaixo do conhecido modelo cliente/servidor.

Para gerência de redes, algumas tarefas de recuperação de informações de roteamento (geralmente uma tabela com várias linhas) existentes no roteador, por exemplo, poderiam ser realizadas utilizando a mesma abordagem, onde o AM selecionaria ou compactaria as informações no destino, e somente depois as enviaria ao gerente.

Figura 3-10: Representação da metodologia de implementação de AM proposta por Dag Johansen.



3.4.3. Sistema de Controle de Tráfego de Veículos

Sistemas de Controle de Tráfego de Veículos são sistemas desenvolvidos com o objetivo principal de informar ao usuário de um veículo uma jornada livre de problemas ao longo de um local A até um local B, selecionando possíveis rotas livres de congestionamentos, incidentes ou barreiras, e ainda possibilitando a mudança de rota em caso de alterações significantes no tráfego.

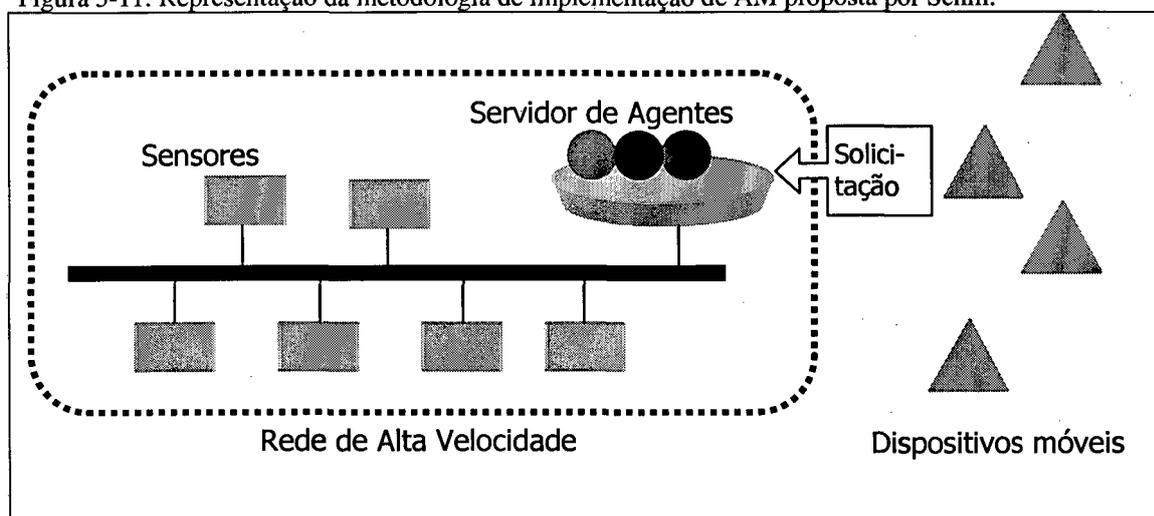
Estes sistemas são formados normalmente por equipamentos com sensores distribuídos por uma área geográfica que é geralmente grande, ligados por uma rede de conexão de alta velocidade; e ainda por dispositivos móveis instalados nos veículos que

geralmente possuem condições mais restritas de comunicação (pequena largura de banda e alta latência⁶).

Para que o sistema funcione, há necessidade da troca de informações de tráfego freqüentemente entre os sensores e os dispositivos móveis. O artigo apresentado em (Schill, 1998) propõe um modelo onde os AM podem ser utilizados para melhorar esta comunicação, diminuindo o tráfego no segmento de rede mais lento.

Como mostra a representação da estrutura da Figura 3-11, os AM seriam iniciados a partir de um dos servidores de agentes disponíveis no sistema (localizados na rede mais rápida) sob solicitação do usuário de um veículo equipado com o dispositivo. Uma vez sendo iniciados, os AM realizariam a coleta de informações entre os sensores envolvidos e processariam qual o melhor caminho a seguir. Como último passo, esta informação seria então repassada ao dispositivo móvel instalado no carro do usuário.

Figura 3-11: Representação da metodologia de implementação de AM proposta por Schill.



Sob este modelo, o tráfego sob o segmento da rede mais crítico (do dispositivo móvel até a rede onde estão interligados os sensores) seria consideravelmente diminuído, pois somente precisaria trafegar nele a solicitação inicial e a resposta obtida, sem a necessidade de que cada sensor fosse consultado diretamente pelo dispositivo.

⁶ Latência ou *network transit delay* é o tempo – medido geralmente milissegundos (ms) – necessário para realizar a transferência de um bit de informação de um ponto a outro de uma rede. A latência é determinada pela largura de banda da conexão, pelo tempo de propagação da rede e também pela velocidade dos equipamentos envolvidos na transmissão. Normalmente, a latência de uma rede *Ethernet* é inferior a 10 milissegundos, enquanto que a latência de uma conexão através de um modem ultrapassa 100 milissegundos (Singhal, 1999).

A utilização de AM neste contexto também traz benefícios quanto ao processamento de várias solicitações em paralelo e à disponibilidade e ao tempo de resposta do sistema, uma vez que todo o processamento passa a ser independente da disponibilidade de equipamentos-chave.

Na Gerência de Redes, este modelo poderia ser utilizado em situações de gerência remota, onde a estação de gerenciamento e os elementos gerenciados estariam separados por um segmento mais lento ou até mesmo por uma conexão via modem.

Uma vez cientes dos conceitos do paradigma de Agentes Móveis, reserva-se ao próximo capítulo, a análise da proposta de seu uso em aplicações de Gerência de Redes.

4. GERÊNCIA DE REDES COM AM

Algumas tarefas dentro da gerência de redes normalmente implicam em um grande volume de informações que dependem além da capacidade e eficiência na análise do administrador da rede ou do software de gerenciamento, da rapidez com que as informações são recolhidas e se tornam disponíveis.

Os modelos de gerenciamento aplicados atualmente (ISO e Internet) são padrões que se baseiam em um sistema simples de troca de informações, de fácil implementação e com um tempo de resposta normalmente rápido, mas que em algumas situações podem consumir recursos que são escassos. Em alguns segmentos de rede podem causar tráfego não desejado na troca de mensagens entre as estações de gerenciamento e os elementos gerenciados.

Desta forma, esforços têm sido realizados para aumentar o nível de distribuição das arquiteturas de gerenciamento mais populares atualmente (SNMP, OSI e TMN) (Wittner, 2000a).

Além das questões de tráfego, o modelo de gerenciamento preferencialmente deve possuir dispositivos que permitam a operação satisfatória mesmo em redes com alta latência. Apesar de simples, os padrões atuais de gerenciamento não possuem qualquer mecanismo para amenizar os problemas gerados na operação neste tipo de situação.

Os AM são uma tecnologia emergente, e configuram-se atualmente como um complemento promissor aos sistemas de arquitetura cliente-servidor (Baldi, 1997). Seu uso na gerência de desempenho têm sido proposto principalmente para a recuperação de informações nos elementos gerenciados e tarefas de alterações de variáveis da MIB (Susilo, 1998), porém, outras áreas da Gerência de Redes podem se beneficiar de seus recursos.

Especificamente, os AM oferecem vantagens potenciais à gerência de redes como: o processamento assíncrono e autônomo de tarefas; a redução de tráfego de rede em determinados tipos de tarefas; a estruturação de um sistema mais robusto e tolerante a falhas; e ainda a possibilidade de processamento otimizado mesmo em redes de alta latência, pois possuem maneiras eficientes de vencer ou minimizar estes efeitos (Schill, 1998) (Lange, 1998). Além disso, devido a sua autonomia em algumas ações, é possível

evitar acessos à rede, processando as informações e tomando as ações cabíveis sem necessidade de consulta à aplicação Gerente, por exemplo.

4.1. Formas de Aplicação Propostas

É praticamente consenso entre os pesquisadores que o uso de AM em aplicações de gerência de redes deve ser uma solução complementar ao paradigma cliente/servidor, atualmente utilizado pelo modelo SNMP. Poucas são as propostas que colocam o uso de AM como uma alternativa ao modelo (Baldi, 1997).

Uma das formas mais discutidas define que a busca e recuperação de informações de gerenciamento na rede deve continuar utilizando uma estrutura onde cada elemento gerenciado possua um servidor de informações (no caso da gerência de redes, a própria MIB ou até mesmo o agente SNMP estacionário), de onde os AM poderiam coletar informações e posteriormente retornar ao usuário (aplicação de gerenciamento ou gerente) os resultados da busca (Braga, 2000a).

Já com relação às tarefas que realizam operações de gerenciamento, os AM podem ser utilizados para alterar listas de roteamento, realizar configurações para garantir o balanceamento de carga e a conectividade de um segmento de rede. Dadas as características dos sistemas de gerência de redes, pode ser desenhado e adotado um sistema com vários agentes, especializados em determinadas tarefas de gerenciamento, responsáveis por toda a rede ou ainda por um determinado segmento desta.

Dentro deste contexto, podem ser aproveitadas várias abordagens de implementação já utilizadas em outros tipos de sistemas: o uso de “servidores de agentes”⁷, esquemas de cooperação e aprendizado dos agentes, estratégias de descarregamento das informações colhidas pelos agentes (Rubinstein, 1999a) (Rubinstein, 2000a) ou ainda existem alternativas que permitem que ações realizadas pelos AM possam ser direcionadas por agentes estacionários, que podem ordenar aos primeiros para que estes se movam para um determinado elemento da rede e lá permaneçam até o término da execução de sua tarefa.

Algumas propostas delegam aos AM até mesmo a tarefa de extrair as informações de gerenciamento diretamente do recurso gerenciado (Bohoris, 2000). Porém, para garantir

⁷ Servidores de Agentes são computadores que contém o código do AM e que, diante de uma requisição feita por outra entidade, podem iniciar a execução do AM, conforme metodologia de implementação vista na Seção 3.4.3 (Schill, 1998).

um tempo de resposta vantajoso, as propostas para o uso de AM em Gerência de Redes necessariamente precisam contar com parte da estrutura no modelo SNMP, metodologia que é adotada na grande maioria dos experimentos realizados pelos pesquisadores (Rubinstein, 1999) (Costa, 1999) (Rivalta, 2000).

Dentro desta última visão, os elementos gerenciados precisam manter agentes estacionários que reuniriam as informações sobre os recursos (MIB) e as quais seriam alvos de solicitações dos AM visitantes, pois como já mencionado, não é viável que o próprio AM visitante realize também esta tarefa.

Pagurek (2000) discorre sobre as formas de abordagem no uso de AM integrados a aplicações de gerência de redes baseadas em SNMP:

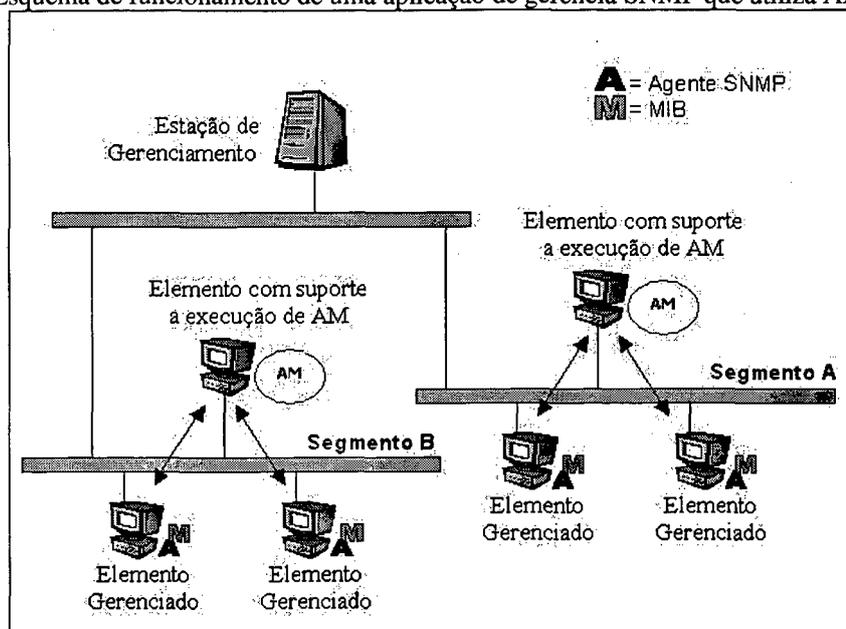
a) os AM poderiam ser providos com a capacidade de agir como um gerente e enviar requisições no formato de mensagens SNMP aos agentes SNMP residentes quando chegassem a um nó, com a vantagem de não necessitar de alterações nos agentes SNMP; poderiam neste caso, também enviar *traps* aos gerentes (Susilo, 1998);

b) os AM poderiam ter a capacidade de estender a MIB, incluindo abaixo de um sub-ramo específico da MIB local, o resultado de suas tarefas para posterior consulta do gerente. Obviamente, deve haver um mecanismo de notificação ao gerente, para que o mesmo possa aproveitar os dados apurados;

c) no caso de um AM ser responsável por um determinado segmento da rede, os agentes SNMP deste segmento podem ser orientados a enviarem *traps* diretamente para o AM que, neste caso, pode agir como um gerente. Nesta abordagem, um agente SNMP deve possuir a habilidade de localizar AM disponíveis na rede, recurso normalmente provido pelas agências do MAS.

Baldi (1997) apresenta um modelo no qual são utilizados vários AM, cada um direcionado a realizar operações em um segmento de rede específico. Uma vez que a estação de gerenciamento inicia os AM, estes são enviados a elementos específicos da rede e de lá, trocam mensagens SNMP com os agentes SNMP localizados em cada elemento gerenciado, conforme é representado na Figura 4-1.

Figura 4-1: Esquema de funcionamento de uma aplicação de gerência SNMP que utiliza AM.



Fonte: Adaptado de (Baldi,1997).

A grande vantagem desta abordagem é que a troca de informações de gerenciamento é realizada localmente (onde geralmente o custo é menor). A partir daí, os AM somente precisariam transferir-se, juntamente com as respostas colhidas, à estação de gerenciamento.

Pagurek (2000) afirma que permitir a comunicação entre AM e agentes SNMP sem garantir um mecanismo de segurança pode representar uma grande fragilidade de todo o sistema de gerenciamento. Assim, sugere que seja provida uma interface segura e controlada, com algum método de autenticação ou até mesmo a adoção de um protocolo intermediário, de forma que AM invasores não possam acessar ou alterar informações no sistema e assim causar situações indesejáveis.

O uso de AM para áreas específicas da gerência de rede – particularmente na gerência de falhas – também é tido como vantajoso e considerado uma ferramenta potencial para desenvolvimento de sistemas de gerenciamento de falhas devido às características de autonomia e mobilidade dos AM, consideradas fundamentais neste tipo de sistema (Wittner, 2000). Uma das formas de aplicação de AM para este tipo de funcionalidade foi comentada na Seção 3.4.1.

A partir de leituras realizadas e da análise de alguns aspectos práticos, pode-se selecionar algumas situações específicas que podem ser especialmente favorecidas pelo uso

de AM. Da mesma forma, pode-se verificar que existem algumas situações onde seu uso seria inócuo ou até mesmo prejudicial, como será visto na Seção 4.3.

4.2. Situações em que o uso de AM poderia apresentar vantagens

A justificativa de redução do tráfego pelo uso de AM somente se aplica em casos onde a comunicação entre o cliente e o servidor não precisa ser intensa, ou seja, onde as **tarefas podem ser executadas de forma autônoma**. É exatamente o caso, dentro da gerência de redes, de algumas das tarefas de monitoração e ajustes no controle da rede.

Na seqüência, principalmente a partir de aspectos práticos, são apresentadas algumas situações em que o uso de AM traria benefício durante o processamento, seja ele economia de banda passante, tempo de resposta ou maior confiabilidade no processamento.

4.2.1. Tarefas que geram um único resultado ou com características de autonomia

É sabido que estas tarefas podem ser executadas através do paradigma cliente/servidor como já foi visto na apresentação do modelo SNMP, mas os AM podem adicionar vantagens no tocante à diminuição da quantidade de dados que trafegam na rede e também quanto ao processamento em paralelo de várias tarefas pelo mesmo servidor.

São exemplos deste tipo de tarefa a “*obtenção da média de pacotes com erros de um conjunto de equipamentos*”, ou a “*descoberta do equipamento que mais envia pacotes com erros à rede acima de um limite aceitável*”, que são tarefas que precisam de uma certa quantidade de consultas a dados distribuídos e de somente um dado como resultado do processamento.

Tarefas sem necessidade de intervenção da aplicação de gerenciamento, por exemplo: a “*mudança ou ajustes na configuração de roteamento no caso de um problema em um dos links de conexão*” também são indicadas para o uso de AM, pois podem ser processadas a partir de regras básicas e simples já embutidas no código do agente. Neste exemplo, seria necessário somente que o agente, após ter realizado o ajuste, notificasse a aplicação de gerenciamento.

Um exemplo interessante é expresso por Baldi (1997), onde os AM são indicados para detectar e corrigir problemas de *loop* no roteamento de uma rede. Ao suspeitar de

alguma anomalia, a aplicação gerente pode enviar um AM, que se transferirá para o roteador mais próximo do problema. Desta localização, o AM usa a informação de roteamento local para mover-se para o local mais próximo do problema, enquanto armazena e compara as informações dos locais visitados, as quais fornecerão condições de detecção do problema.

Obviamente, a própria estação de gerenciamento poderia detectar estes problemas pela leitura das tabelas de roteamento, mas esta leitura poderia ser prejudicada pelo fato que a estação de gerenciamento – devido ao problema de roteamento – pode não ter condições de acessar estas informações, uma vez que informações parciais de roteamento não são suficientes para detectar o problema.

4.2.2. Redes com alta latência ou gargalos

Algumas tarefas de controle podem sofrer atrasos consideráveis dentro do modelo cliente/servidor caso a rede possua segmentos com tempos de latência mais altos entre a estação de gerenciamento e o elemento a ser gerenciado.

Dada a necessidade de reações que precisam ser aplicadas em caráter emergencial em um determinado elemento da rede para prevenir ou contornar um problema, os AM podem ser enviados de pontos mais próximos que não precisam submeter-se a este segmento mais lento para a realização da tarefa.

Nesta situação, até mesmo tarefas simples, como a recuperação de informações em cada um dos elementos do outro lado do segmento lento também podem se beneficiar do uso de AM. Os dados de resposta, que seriam enviados à estação de gerenciamento através do segmento mais lento, poderiam ser previamente processados ou compactados pelo AM, de forma a fazer com que o menor número possível de informações de gerenciamento trafegue por este segmento problemático.

Situação semelhante acontece quando a estação de gerenciamento encontra-se conectada à rede onde estão os elementos a serem gerenciados por uma conexão lenta e com alto custo (gargalo). O tráfego gerado pelo AM nesta conexão seria muito inferior ao gerado pelo modelo SNMP, pois uma vez que tenha chegado ao segmento que contém os elementos à gerenciar, precisaria passar por ele novamente somente ao término da tarefa.

Da mesma forma, a gerência de redes com segmentos de rede com altas taxas de perda de pacotes pode ser beneficiada com esta mesma modelagem, proporcionando assim melhor tolerância a falhas de transmissão.

4.2.3. Segmentos de Conexão Instável ou Temporária

Os AM podem ser úteis em redes que possuam segmentos que não são permanentemente conectados à ela, e onde existam elementos a serem gerenciados. Uma vez enviados para este segmento, independentemente de que a conexão esteja ativa, o AM poderia processar a tarefa. Estando o segmento novamente conectado, o AM enviaria os resultados colhidos ou processados à aplicação de gerenciamento.

Normalmente, a gerência de redes tradicional via SNMP seria muito prejudicada nesta situação, pois não seria viável realizar requisições sucessivas a um segmento que não responde e mesmo quando recebidas nos elementos gerenciados, as respostas emitidas pelos agentes SNMP também seriam perdidas quando da ocasião de nova queda da conexão no segmento.

4.2.4. Aplicações alternativas de gerência em equipamentos portáteis

Em determinadas situações, os gerentes de rede poderiam ter à sua disposição equipamentos portáteis, como *notebooks* ou *palmtops* que poderiam exibir informações de gerenciamento em ocasiões em que o gerente não estivesse fisicamente na organização.

Estes equipamentos ficam a maior parte do tempo desconectados da rede e possuem normalmente conexões frágeis e sujeitas a freqüentes quedas. Para recuperar informações ou executar operações de modificação na configuração da rede, o gerente poderia enviar um AM a uma central fixa da rede, que iniciaria a execução da tarefa, e uma vez estando a tarefa concluída, armazenaria o resultado. O gerente, ao se conectar, teria então acesso à informação conseguida ou ainda, ao resultado da tarefa que solicitou ao AM, sem a necessidade de permanecer conectado durante esta execução.

4.2.5. Situações de rede desfavoráveis ou imprevistas

Graças também à propriedade de adaptação, os AM podem reagir dinamicamente diante de situações desfavoráveis ou eventos imprevistos, o que facilita a modelagem e construção de um sistema de gerência de redes mais tolerante a falhas.

No caso de problemas na estação de gerenciamento ou em um dos elementos gerenciados constantes em seu itinerário, o AM pode continuar executando sua tarefa independentemente deles. Inclusive, podem ser implementados dispositivos para que a agência do MAS informe aos agentes hospedados que o elemento gerenciado não está disponível, de forma que os AM são dinamicamente moldados para que suas tarefas reflitam o mínimo possível este problema.

O sistema pode inclusive notar a falta de um AM - no caso de um problema no equipamento que o mesmo estava hospedado apresentar falhas - e o mais breve possível tomar providências para que sua ausência seja corrigida (Gudwin, 2000).

4.2.6. Tarefas com operações de alterações uniformes

Talvez o maior ganho no uso de AM com relação à economia de mensagens trocadas entre os elementos gerenciados e a estação de gerenciamento seja na realização de tarefas de alteração nas variáveis da MIB de um conjunto de elementos gerenciados.

A partir do envio ou da solicitação inicial para que o AM realize a tarefa de alteração, o mesmo pode viajar por todos os elementos da rede realizando operações SET sobre variáveis da MIB sem precisar manter contato constante com a estação de gerenciamento.

São operações com este comportamento, por exemplo, ajustes no tamanho do maior datagrama IP aceito pelos elementos da rede (variável *ifMtu*), número máximo de conexões TCP que as entidades podem sustentar (variável *tcpMaxconn*), a máscara de sub-rede para roteamento IP (variável *ipRouteMask*), a adição de um endereço na tabela de roteamento (*ipRouteTable*), ou ainda, algum ajuste em variáveis específicas de um tipo de equipamento, por exemplo, configurar todas as impressoras para imprimirem em modo econômico.

Nestes exemplos, o AM apenas precisaria ao final de sua tarefa, reportar à estação de gerenciamento se a mesma foi executada com sucesso ou não.

4.3. Situações em que o uso de AM poderia apresentar desvantagens

Segundo Braga (2000), “existe uma tendência de utilizar novas tecnologias como solução para a maioria dos problemas, porém, existem vários tipos de aplicações que não são adequadas a este paradigma. Várias justificativas para o uso de AM não procedem, senão como teste de uma nova técnica, não trazendo maior eficiência na execução de algo que sistemas anteriores já o faziam”.

A princípio, não teria sentido, por exemplo, utilizar agentes móveis para a obtenção de informações SNMP em um número reduzido de elementos de rede de um segmento de rede uniforme, ou seja, sem nenhum tipo de gargalo, pois em uma análise rápida e superficial, seria possível deduzir que o tráfego de rede das informações recuperadas na rede seria somado ao código do AM que também viajaria no mesmo segmento, gerando mais tráfego de rede que as mensagens SNMP, geralmente bem menores (Baldi, 1997).

Pode-se admitir, porém, que exista uma razão com relação ao número de elementos a serem gerenciados, a partir da qual o uso de AM passa a ser mais vantajoso que a gerência SNMP tradicional. Esta razão (aplicada aos protótipos desenvolvidos) pode ser conhecida a partir da análise dos resultados obtidos nos experimentos, que é apresentada no Capítulo 7.

Normalmente, sistemas distribuídos demandam protocolos que envolvem múltiplas interações para concluir uma tarefa, porém, este não é o caso do modelo de gerência SNMP, que é baseado no protocolo SNMP funcionando sobre o protocolo UDP, que não é orientado à conexão⁸ (Stallings, 1999). Considerando este aspecto, os AM não implicam necessariamente em redução significativa do tráfego de rede nas tarefas normais de monitoramento dentro da gerência de redes, o que é facilmente confirmado em outros tipos de sistemas (Gudwin, 2000).

Entretanto, é possível aplicar algumas otimizações no desenho de aplicações de Gerência utilizando AM que poderiam ser úteis neste sentido, como é discutido na Seção 4.4 e diante de sua aplicação nos experimentos, seus benefícios são apurados no Capítulo 7.

⁸ Em um serviço não orientado à conexão, a transmissão acontece sem que haja necessidade de estabelecer uma conexão entre emissor e receptor, caso contrário, é necessária a troca prévia de mensagens para que a conexão seja estabelecida e então, a transmissão seja iniciada.

4.3.1. Elementos gerenciados com recursos de execução restritos

O melhor desempenho de execução dos AM depende diretamente dos recursos disponíveis em cada elemento a ser gerenciado (Costa, 1999). Assim, se um elemento não dispõe de recursos de memória e processamento satisfatórios, a execução do AM está relativamente comprometida.

Da mesma forma, se a rede a ser gerenciada possui estas características na maioria de seus elementos, a Gerência de Redes através de AM pode ter graves implicações e até mesmo se tornar inviável. Porém, observando alguns aspectos durante a ocasião da implementação do AM, é possível minimizar estes efeitos através da propriedade da *Adaptação*, uma das características de qualquer agente de software, já mencionada no Capítulo 3.

Conceitualmente, um agente de software pode ter a habilidade de adaptar-se dinamicamente através do sensoriamento do ambiente de execução (taxa de ocupação de memória e CPU, ociosidade, entre outros) e a partir daí, reagir autonomamente de forma a conseguir o melhor tempo de processamento possível naquele equipamento.

Falando particularmente de Java, a linguagem não provê mecanismos eficientes em sua máquina virtual para limitar os recursos utilizados por um objeto específico (Lange, 1999). Portanto, para evitar o uso indiscriminado de recursos, deve-se tomar as precauções diretamente no código do AM, como é feito, por exemplo, no ambiente MOLE (Baumann, 1999).

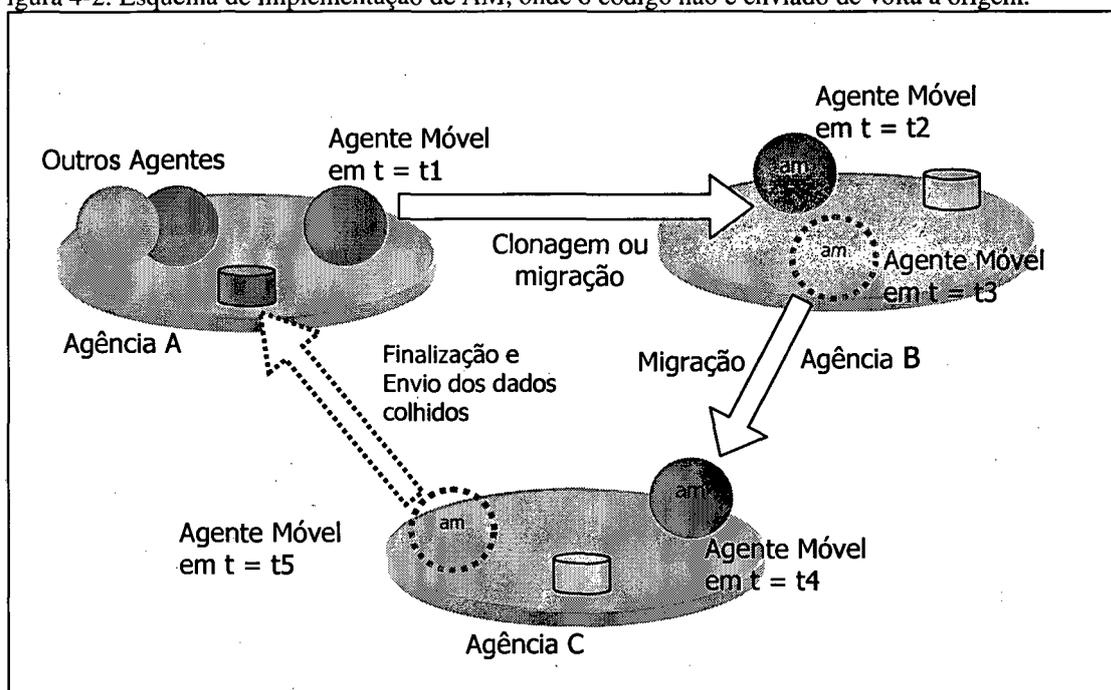
4.4. Desenvolvimento de aplicações com AM para Gerência de Redes

Vários experimentos (Rubinstein, 1999) (Costa, 1999) (Braga, 2000) caracterizam protótipos de AM como códigos que possuem o seguinte comportamento: podem transferir-se para cada ponto da rede, buscar alguma informação, armazená-la e depois se transferir novamente para o local onde foi criado e ativado. Esta é a abordagem mais simples e conhecida.

Na verdade, a questão do retorno do AM ao local de origem não é uma exigência no desenvolvimento de sistemas de agentes, pois um AM pode, ao final de sua tarefa, apenas enviar à origem os dados colhidos, sem a necessidade de enviar seu código junto. Tal

abordagem torna-se interessante principalmente quando existe a preocupação de economia de recursos de rede, em segmentos mais lentos, com largura de banda mais baixa ou mesmo quando existe a preocupação que a aplicação de gerência cause mínima interferência na operação normal da rede. A Figura 4-2 representa este método.

Figura 4-2: Esquema de implementação de AM, onde o código não é enviado de volta à origem.

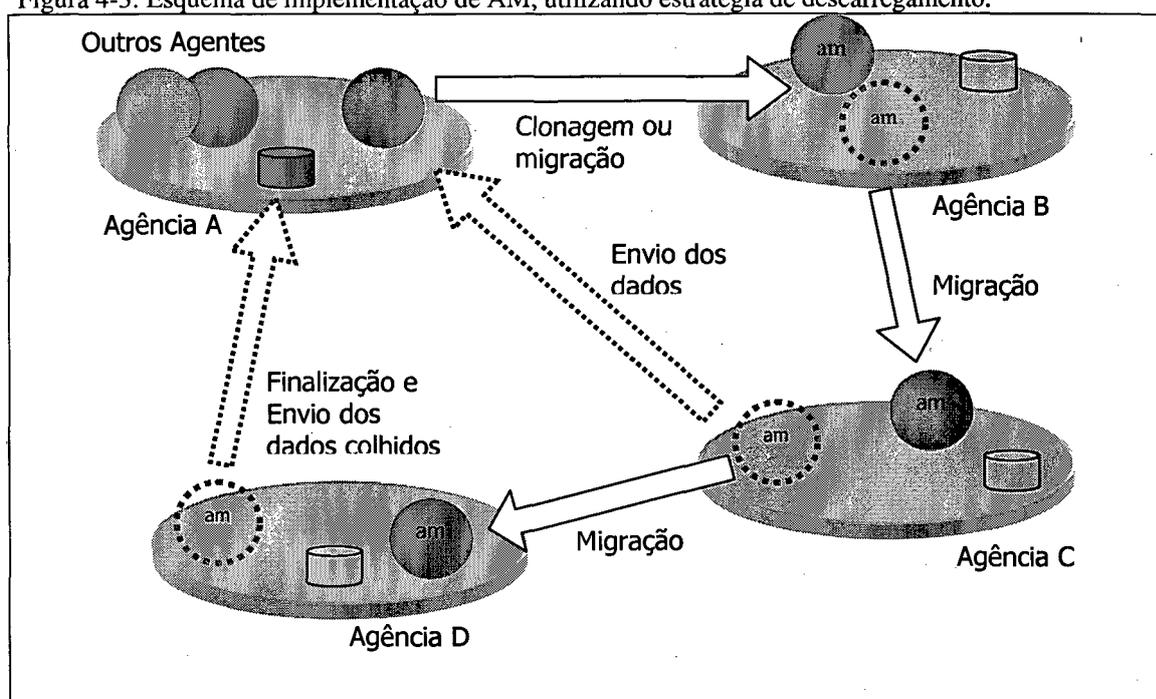


Mais recentemente, Rubinstein (2000) apresenta estudos que validam estratégias para reduzir o tempo de resposta, as quais chama de “estratégias de descarregamento”: primeiramente, não retornar à origem (apenas mandar os dados coletados à estação de gerenciamento – como já foi mencionado); e posteriormente permitir que um AM possa ser programado para enviar os dados coletados à Estação de Gerenciamento quando atinge uma determinada quantidade de elementos gerenciados visitados (Figura 4-3), evitando assim que a transferência de informações durante o trajeto do AM não se acumule, se tornando um fator que possa prejudicar o desempenho ou gerar condições favoráveis ao aparecimento de erros de transmissão.

Rubinstein inclusive verificou por meio de simulações que estas estratégias podem reduzir significativamente o tempo de resposta do AM com relação à sua abordagem mais tradicional (vai-volta) e em razões mais modestas, até mesmo em relação ao SNMP (Rubinstein, 2000) (Rubinstein, 2000a). Neste último estudo, foram apuradas estimativas e

realizadas análises do uso de recursos da rede (tráfego) gerado por um dos modelos de implementação: que utiliza múltiplos agentes simultaneamente para a realização da tarefa. Foi constatado que o uso de múltiplos AM, apesar de significar uma diminuição no tempo de resposta, representa também uma utilização maior de recursos da rede em relação ao SNMP. As simulações, apesar de considerarem que os nós da rede não possuem carga e que os *links* são livres de erros, confirmam que o uso de um único AM pode reduzir drasticamente o tráfego de informações de gerenciamento em relação ao SNMP, a partir de um número específico de elementos gerenciados.

Figura 4-3: Esquema de implementação de AM, utilizando estratégia de descarregamento.



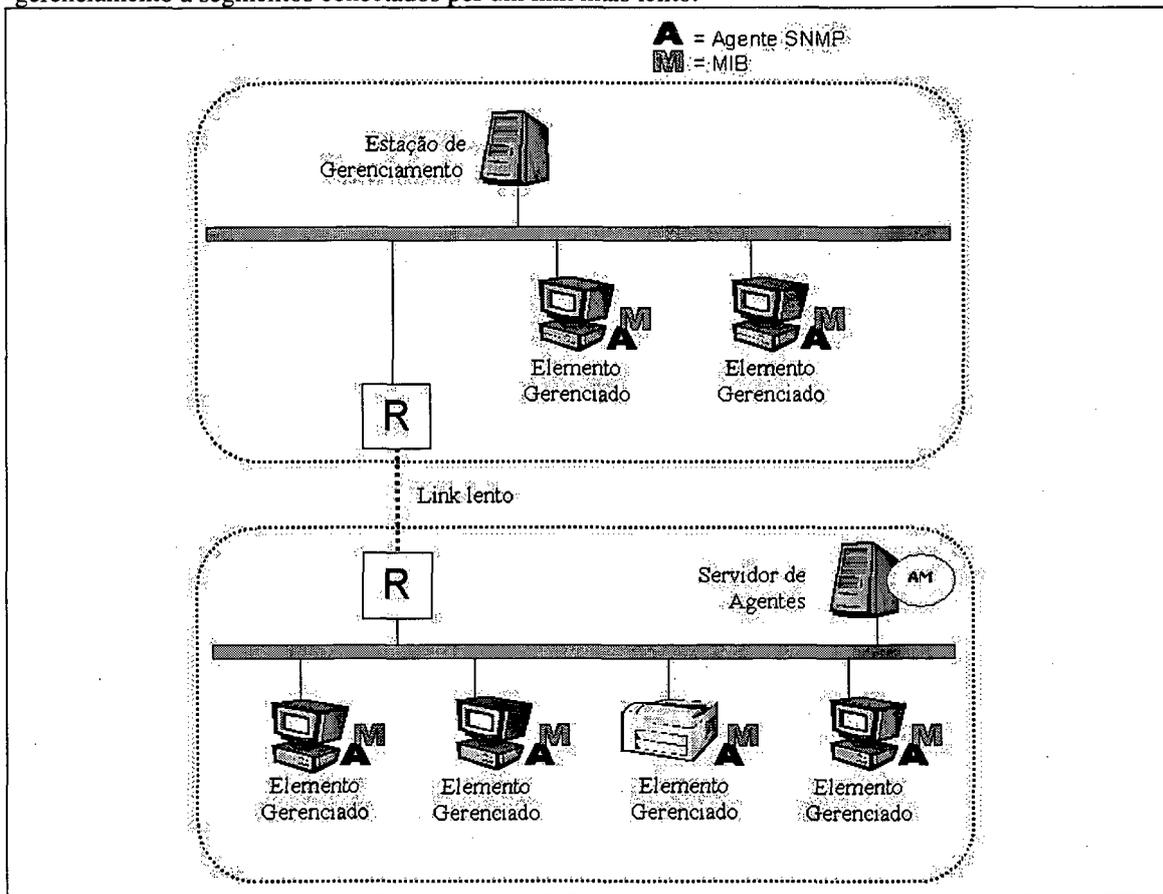
Com o objetivo de poupar um segmento de rede mais lento (gargalo), o presente trabalho sugere que é interessante implementar um sistema com “servidores de agentes”, onde, no momento em que a estação de gerenciamento precisa de alguma informação, uma solicitação é enviada ao servidor de agentes, que estando o mesmo do outro lado do segmento, iniciaria o AM para a realização da tarefa solicitada.

Após sua execução, o AM então enviaria as informações colhidas à estação de gerenciamento. Assim, somente a solicitação inicial e os dados colhidos trafegariam pelo segmento problemático (Figura 4-4). A mesma lógica pode ser aplicada quando a estação de gerenciamento envia a solicitação diretamente ao AM, que estaria confinado àquele segmento, sendo o resultado de sua tarefa enviado por ele de volta à estação.

Com relação à detalhes de implementação, no uso de metodologias em que, após a inicialização, o AM leva consigo seu itinerário, é interessante implementar mecanismos em que na medida que o elemento gerenciado tenha sido visitado e seus dados forem enviados à estação de gerenciamento, o vetor que armazena os endereços de itinerário seja revisado de modo que os endereços já visitados sejam excluídos e assim não mais transferidos junto com o AM.

Com relação à infra-estrutura do sistema, alguns ambientes para desenvolvimento de AM como o Aglets Workbench da IBM (Lange, 1998a), possuem mecanismos que permitem o armazenamento de classes para uso posterior, onde, uma vez que o AM foi executado em um elemento de rede, todas as suas classes são armazenadas nele, e desta forma, somente há necessidade de nova transferência de classes quando houver alterações no código. É possível conseguir melhores tempos de resposta na execução a partir da segunda vez que o AM é executado em um conjunto de equipamentos, além da redução do tráfego gerado.

Figura 4-4: Implementação de AM que utiliza servidores de agentes para otimizar operações de gerenciamento a segmentos conectados por um link mais lento.



Com o objetivo de contribuir para a comprovação das vantagens do uso de AM para as atividades de Gerência de Redes, a implementação de cinco protótipos foi realizada. Além dos protótipos SNMP e do AM tradicional (vai-volta), foram implementados outros três protótipos de AM, que contemplam as otimizações descritas acima (enviar somente resultado, envio do resultado em intervalos e utilizando servidor de agente). Tais protótipos foram utilizados para realização dos experimentos (cujas metodologias são descritas no capítulo 6), de modo a verificar a validade destas alternativas em apresentar maiores benefícios.

São apresentadas na seqüência, as ferramentas que foram utilizadas na implementação e as particularidades dos protótipos desenvolvidos para os experimentos.

5. AMBIENTE DE DESENVOLVIMENTO

O sistema foi implementado utilizando os recursos da orientação a objetos, visto que a maioria dos sistemas de AM utiliza este paradigma.

A linguagem de programação escolhida para implementação dos protótipos foi Java (versão 2), disponível⁹ através do kit da Sun (JDK 1.3), por reunir condições e ferramentas que agilizam o desenvolvimento dos protótipos, tanto de SNMP quanto de AM, e pelo fato de concretizar-se como uma tecnologia emergente que pode prover a base para a resolução de problemas com sistemas de gerenciamento legados. Além disso, tem condições de ser a opção adotada para implementação nativa em dispositivos de rede (White, 1998).

Considerando que pode haver influência sobre o tempo de resposta (Rubinstein, 1999) (Rubinstein, 2001), foram definidos que ambos os protótipos devem adotar a mesma versão e linguagem de desenvolvimento, de forma que a comparação seja feita da forma mais transparente e em igualdade de condições possível.

Com relação às funcionalidades do SNMP, foi utilizado um conjunto de classes chamado *Java Dynamic Management Kit* (JDMK), cuja estrutura será mostrada neste capítulo. Para o desenvolvimento dos protótipos de AM, inicialmente foi escolhida a plataforma MCT (*Mobile Code Toolkit*) por ter seu código aberto, além de ter sido desenvolvida na Universidade de Carleton (Canadá) onde se encontram vários pesquisadores que figuram como referência em praticamente todos os artigos que envolvem o assunto, como: Pagurek, White, Schramm e Bieszczad.

No entanto, erros no código fonte das classes disponíveis nas páginas do projeto na Internet, talvez devido a alterações feitas por diferentes membros do projeto ao longo dos seus cinco anos de trabalho, não permitiram que o protótipo tivesse seu funcionamento estável, apresentando erros frequentes de execução, o que levou a adoção da plataforma *IBM Aglets Workbench*, mostrada também neste capítulo. Com esta escolha, os protótipos foram implementados sem que qualquer falha nas classes ocorresse e tornou os protótipos com AM também compatíveis com a versão 2 do Java, o que não acontecia com a plataforma anterior.

⁹ Disponível *on-line* no endereço: <http://www.sun.com>

5.1. Sun Java Development Kit 1.3

O Java pode ser considerado mais que uma linguagem de programação, trata-se de um padrão de implementação que inclui uma rica hierarquia de classes para comunicação em redes TCP/IP (White, 1998). Possui facilidades que permitem a implementação de técnicas inovadoras de gerenciamento baseadas na mobilidade de código, provendo inclusive, um ambiente de execução independente de plataforma.

Com relação às questões de migração, segundo Fünfroeken (1998), o Java somente suporta a captura do estado das variáveis do programa, processo este conhecido como *serialização*, porém, não provê mecanismos para a captura da pilha de processos ou o contador do programa em execução na máquina virtual Java tradicional (Suri, 2000).

Porém, este fator não se torna limitador para o uso de Java no desenvolvimento de AM, visto que foram desenvolvidos mecanismos para captura do estado de um programa Java em nível de linguagem. Estes mecanismos alternativos implementam o armazenamento e restauração das informações de estado, de forma que algumas funcionalidades deste tipo de mobilidade sejam garantidas não pela linguagem em si, mas por comandos inseridos em seu código.

Como vantagens do uso de Java em aplicações de AM, Milojevic (1999a) cita: suporte ao código móvel, heterogeneidade, serialização de objetos, carregamento dinâmico de classes e *multi-threading*. Porém, o suporte ao gerenciamento de recursos locais (memória, cpu) é precário, além de não haver suporte nativo para obter informações do contexto de execução de uma *thread* (Lange, 1998a).

5.2. Java Dynamic Management Kit 4.2

Para construção dos protótipos, foi utilizado o *Java Dynamic Management Kit*, que possui classes que permitem implementar uma estrutura de gerenciamento SNMP (gerentes, MIB e agentes).

O conjunto de classes e ferramentas conhecido como JDMK - desenvolvido pela *Sun Microsystems* - é uma especialização de outro pacote lançado anteriormente: o JMX (*Java Management Extensions*) e oferece instrumentação para criação de sistemas e aplicações de gerenciamento de redes, incluindo um conjunto de classes e métodos que permitem a

criação e a manipulação de agentes estacionários, gerentes, MIB's SNMP e suas interações, via mensagens SNMP.

Além disso, oferece a possibilidade de criação de interfaces para *web* que permite que as aplicações de gerenciamento desenvolvidas possam ser executadas diretamente em navegadores, porém, esta funcionalidade não foi utilizada nos protótipos desenvolvidos.

Os componentes de programação do JDMK são: a arquitetura para implementação da especificação JMX, os módulos de comunicação (para acessar os agentes SNMP remotamente), serviços para os agentes (monitoramento, hierarquia, localização e segurança) e serviços SNMP (que permitem a integração dos agentes desenvolvidos com aplicações padronizadas já existentes) (Sun, 2000).

Será visto agora o subconjunto de objetos JDMK utilizados para desenvolvimento dos protótipos necessários aos experimentos.

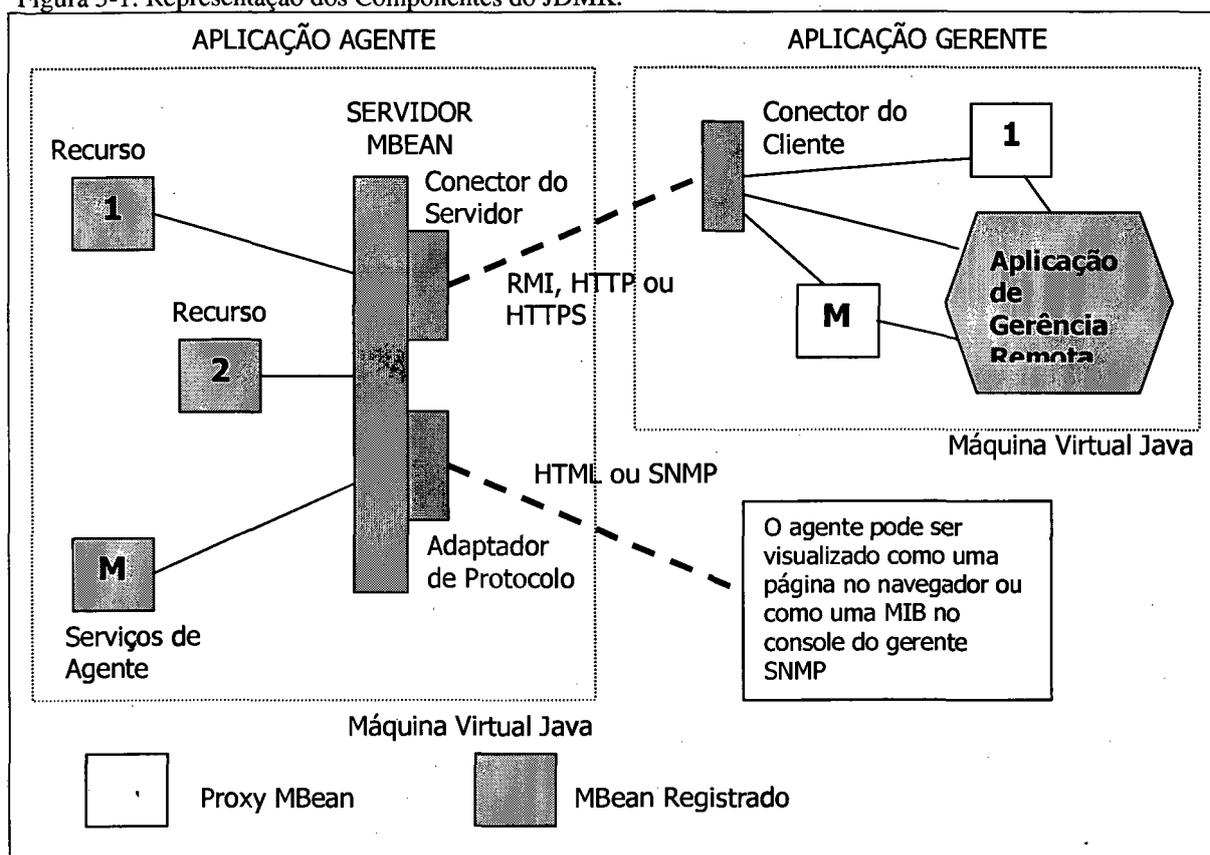
5.2.1. Componentes

Existem cinco componentes que permitem definir o gerenciamento de rede utilizando o conjunto de classes JDMK: servidor *MBean*, *MBeans*, adaptadores, conectores e *proxy MBeans*. A representação destes componentes pode ser vista na Figura 5-1.

Um servidor *MBean* atua como um registro para os recursos de gerenciamento e permite que sejam acessadas as operações disponíveis para estes recursos, através de sua interface (Mello, 2000). Assim, pode-se criar e gerenciar recursos a partir dele.

Um *MBean (Managed Bean)* é um objeto Java em conformidade com padrões de especificação derivados do modelo de componentes *JavaBeans*. Um *MBean* trabalha com padrões de especificação para expor atributos e operações, que habilitam qualquer agente JDMK a reconhecer e gerenciar o *MBean*. Um *Mbean* pode representar qualquer entidade, seja ela física ou virtual. Através dele, pode-se acessar os atributos e operações disponíveis àquela entidade.

Figura 5-1: Representação dos Componentes do JDMK.



Fonte: (Sun, 2000).

Um *proxyMBean* é a representação de um objeto *MBean*. Assim, aplicações gerentes podem acessar recursos (buscar ou alterar atributos, executar operações e receber notificações) através da interface *proxyMBean*.

Conectores permitem que as aplicações de gerenciamento comuniquem-se com o servidor *MBean*. Esta conexão pode ser realizada através de protocolos: HTTP (*HiperText Transfer Protocol*), TCP ou ainda, RMI (*Remote Method Invocation*).

Assim, para que o servidor *MBean* e a aplicação de gerenciamento se comuniquem, é preciso que o código da aplicação declare e instancie uma classe "conector", que será o servidor de conexão. Da mesma forma, o gerente deve realizar a mesma operação, transformando-o em "cliente de conexão".

Adaptadores de protocolo realizam a conexão entre os elementos da aplicação de gerenciamento e devem utilizar um protocolo específico: SNMP, parâmetro que é informado na ocasião de sua criação.

Os conectores e adaptadores permitem ainda, que uma aplicação de gerenciamento consiga localizar os MBeans existentes através de seus nomes, buscar ou acrescentar atributos às instâncias de MBeans existentes, instanciar e registrar novos MBeans no servidor de MBeans e receber notificações de MBeans já criados.

A partir destes cinco componentes, pode-se então, iniciar a construção da estrutura de gerenciamento SNMP. Para tanto, foram utilizados como referência os documentos Sun (2000, 2000a e 2000b).

5.2.2. O desenvolvimento do Protótipo SNMP

Para criação de uma aplicação de gerenciamento SNMP, foi necessária primeiramente a criação de uma MIB no formato requerido pela ferramenta: um conjunto de *MBeans*. Para todos os protótipos foi adotada a MIB padrão RFC-1213 (MIB-2).

O pacote JDMK possui um aplicativo que realiza esta tarefa: MIBGEN. A partir da descrição textual da MIB, o MIBGEN cria e instancia, no servidor *MBean* local, o conjunto de todos os *MBeans* que irão corresponder exatamente a cada ramo da MIB, cada qual contendo seus atributos.

Além disso, este aplicativo cria os métodos *get* e *set* para cada um dos *MBeans* criados, para respectivamente, recuperar e alterar o valor destas variáveis. Uma vez criadas as classes Java que correspondem a MIB, estas devem ser utilizadas diretamente no código dos agentes e gerentes a serem desenvolvidos.

Porém, estas classes necessitam de implementação adicional para que, uma vez instanciadas, contenham valores que reflitam as condições reais do equipamento ou do ambiente de execução, por exemplo, para que o ramo *sysDescr* contenha a descrição do sistema, é necessária a implementação de linhas de código que descubram este valor a partir do equipamento em que a MIB estiver sendo executada. Desta forma, estas linhas de código foram adicionadas ao código das classes geradas.

Em um segundo momento, passou-se para o desenvolvimento do agente SNMP: nele, foram embutidas linhas de código para a criação do servidor *MBean*, definição do domínio de gerenciamento sob o qual o agente será estabelecido e definição do adaptador

SNMP, para o qual foi utilizada a porta¹⁰ 8085. Normalmente, aplicações de gerência SNMP utilizam a porta 161, a qual não foi utilizada para prevenir interferência de outras aplicações existentes na rede a ser testada.

Ainda no código fonte do agente SNMP, foram incluídas instruções para instanciação da MIB, seu registro do servidor MBean e formas para que o agente agisse como um programa residente. Assim, o agente SNMP foi concluído e está apto para responder requisições via protocolo SNMP. Para que os experimentos fossem realizados, em cada elemento gerenciado foi instalado e executado o agente SNMP.

Quanto à aplicação de gerenciamento, a mesma primeiramente localiza todos os agentes SNMP ativos na rede. A partir da lista conseguida, envia uma mensagem no padrão do protocolo SNMP para todos os agentes estacionários, solicitando, por exemplo, as variáveis *sysDescr* ou *ifInErrors*. Esta aplicação contabiliza o tempo de resposta da operação, que começa a ser contado a partir do envio da primeira mensagem de solicitação. O tempo de resposta é obtido quando acontece o recebimento da mensagem de resposta do último agente SNMP consultado.

Todas as solicitações de variáveis, montadas como mensagens SNMP, são enviadas seguidamente sem que seja necessário o recebimento das respectivas respostas, assim, todo o processamento é realizado de forma assíncrona, como define o modelo SNMP. Quando a última mensagem é recebida, o tempo total de realização da tarefa é computado e mostrado também na tela.

O procedimento para a marcação do tempo de resposta adotado foi o mesmo utilizado em (Fiorese, 2000) que diminui o tempo final do inicial, ambos obtidos em milissegundos dentro do código em Java.

5.2.3. Dificuldades encontradas

Devido à preocupação de cumprir o cronograma de realização deste trabalho e a grande quantidade de elementos, nem todos os ramos da MIB puderam ser implementados, assim, as classes que não puderam ser implementadas possuem em seu código um comando

¹⁰ Portas são endereços que representam processos em um *host*. Cada um dos processos em execução deve possuir um número único de porta no *host*, o que irá permitir que os dados sejam entregues diretamente ao processo apropriado.

que inicia seus atributos com valores simulados, ou seja, não foram obtidas diretamente do ambiente de execução.

Foram implementados os seguintes ramos da MIB: *System*, *Interfaces* e *Snmp*, porém, nem todas as variáveis do ramo *Interfaces* possuem valores reais. Estas variáveis foram iniciadas com valores aleatórios com o objetivo de simulação, já que os experimentos somente se preocupam em recuperar a informação destas variáveis mesmo que sejam simuladas.

Uma decisão tomada foi o desenvolvimento do protótipo sem a utilização da interface gráfica, para evitar que o tempo de manipulação de objetos da interface fosse computado no tempo de resposta obtido nos experimentos. Assim, todos os resultados e mensagens tanto do agente quanto do gerente SNMP são mostrados em modo texto. Esta forma foi adotada principalmente porque os protótipos construídos para os experimentos com AM também não manipulam a interface gráfica em seu código.

5.3. IBM Aglets Workbench 1.2

Aglets Workbench é uma plataforma para o desenvolvimento de AM, criada pela IBM do Japão. Atualmente, esta plataforma encontra-se na versão 1.2, desenvolvida para suportar Java 2. A palavra *Aglet* significa “*Agent and Applet*”, pois teve inspiração no modelo de *Applets*¹¹, desenvolvido pela Sun (Lange, 1998a).

Sendo a plataforma pioneira para o desenvolvimento de AM (Bellavista, 2001), todo o código desenvolvido foi aberto pela IBM, tratando-se a partir de então de uma plataforma “*open source*”, com o objetivo principal de atrair interessados em implementar novas funcionalidades e corrigir alguns problemas conhecidos na adaptação com nova versão do Java, em adição à versão anterior (1.1), a qual suportava somente Java 1.

Aglets são objetos Java que podem mover-se de um lugar para outro na rede. Uma vez que esteja sendo executado em um determinado elemento, o *Aglet* pode interromper sua própria execução e despachar-se para outro elemento da rede à sua escolha, onde reiniciará sua execução. Durante esta operação, o *Aglet* carrega consigo seu código de programa e seu estado.

¹¹ *Applets* são aplicativos gerados a partir de classes JAVA que têm a característica de se transferirem do servidor para o navegador do usuário, de onde iniciam sua execução (Lange, 1998a).

A plataforma *Aglets Workbench* provê ainda um mecanismo de segurança com o objetivo de garantir a segurança de recepção e execução do código de agentes remotos, ou seja, viabilizar a segurança pela verificação do conjunto de bytes recebidos pode impedir o uso de seqüências de código proibidas. Existe ainda o mecanismo de segurança em nível de comandos, que pode definir quais recursos um programa Java pode acessar (Gudwin, 2000).

Além disso, é um modelo para a programação de AM que suporta mecanismos de comunicação entre agentes através de um modelo de mensagens, que inclusive podem ser trocadas com agentes estrangeiros à plataforma, pois este serviço segue rigorosamente o padrão *MASIF* (comentado na Seção 3.1).

O ambiente dentro da plataforma *Aglets* foi desenvolvido de forma que a máquina virtual Java não precisasse sofrer quaisquer alterações, simplesmente agregando a ela uma API¹² (*Application Programming Interface*) com classes que permitem toda a operação (Lange, 1998a).

5.3.1. Componentes

A plataforma *Aglets* é formada por um aplicativo que realiza a função de ambiente de execução de agentes chamado TAHITI. O TAHITI trabalha realizando as funções de uma agência, vistas na Seção 3.1. Para cada equipamento que se deseja incluir no MAS, obrigatoriamente deve-se executar o aplicativo TAHITI, que usa a porta número 4434 para realizar as operações com AM, mas que pode ser alterada.

A implementação do TAHITI utiliza o protocolo ATP (*Agent Transfer Protocol*), um protocolo em nível de aplicação utilizado em sistemas baseados em agentes distribuídos, e possui os seguintes métodos padrões para realizar operações com AM (Gudwin, 2000):

a) *dispatch()*: sua função é requisitar à agência destino que receba o agente enviado no conteúdo da requisição e inicie sua execução; após esta operação, se tudo for realizado com sucesso, a agência de origem finaliza a execução do agente e libera os recursos que este estava utilizando;

¹² Dá-se o nome de API ao conjunto de rotinas que uma aplicação usa para solicitar e fornecer serviços de baixo nível ao sistema operacional (Starlin, 1998).

b) *retract()*: envia uma solicitação à agência destino que retorne o agente especificado, enviado à ela anteriormente por uma operação *dispatch*;

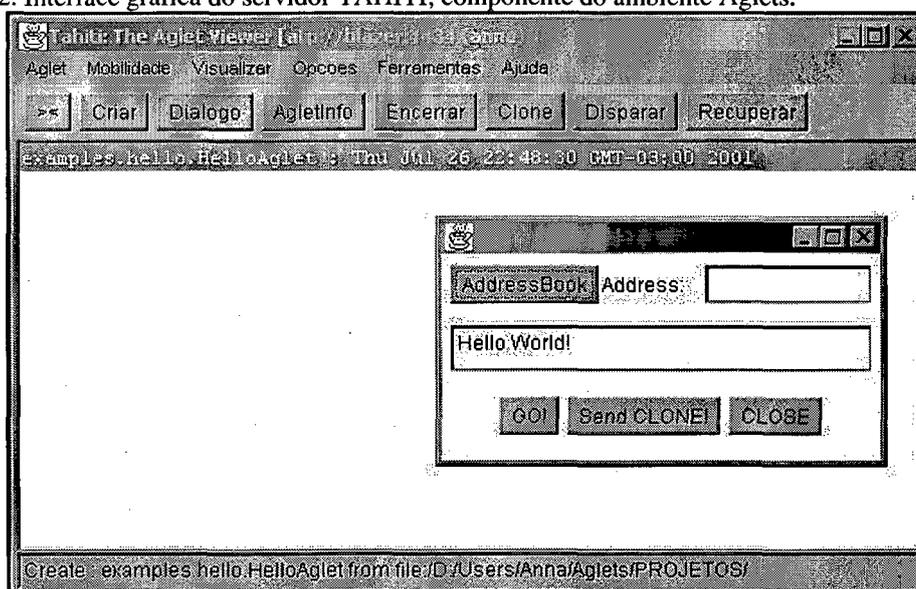
c) *create()*: cria o *aglet* conforme as classes especificadas pelo proprietário, ou seja, inclui o *Aglet* especificado na agência, o qual recebe um identificador único;

d) *dispose()*: finaliza a execução de um *Aglet*;

e) *cloning()*: cria um clone do *Aglet* especificado. Este clone pode ser criado tanto na agência local quanto em uma outra agência conhecida.

O TAHITI possui uma interface gráfica que permite ao administrador monitorar e controlar Aglets, possuindo botões que podem executar as principais funções de uma agência: criar, enviar, mostrar informações de identificação, finalizar e recuperar agentes. A interface gráfica pode ser vista na Figura 5-2.

Figura 5-2: Interface gráfica do servidor TAHITI, componente do ambiente Aglets.



Cada linha desta tela representa um AM, que pode, por exemplo, ser enviado para outro elemento da rede que esteja executando o TAHITI. A implementação do AM pode permitir que seja informada uma relação de endereços a visitar – o itinerário. Para enviar manualmente um agente, existem os botões de controle que também podem ser vistos nesta tela, porém, cada agente pode ser implementado de forma que o próprio defina o momento que irá iniciar sua execução e até mesmo quais endereços irá visitar.

O ambiente de desenvolvimento Aglets possui em sua biblioteca várias soluções e classes, as quais, dão condições para criação de um MAS de acordo com a necessidade da aplicação (Lange, 1999). Alguns componentes do ambiente Aglets são:

- Classe *Aglet*: é uma classe abstrata que define métodos fundamentais para AM, onde todo AM desenvolvido deve estender (herdar métodos, variáveis e interfaces). À classe *Aglet* pertence a variável *AgletInfo*, que armazena atributos e informações sobre o criador e a agência onde o *Aglet* foi criado.

- Interface *AgletProxy*: sua principal finalidade é viabilizar a comunicação entre *Aglets*, que sempre são acessados por meio de seu *proxy* e nunca diretamente.

- Classe *Message*: utilizada na comunicação entre *Aglets*, é a classe que implementa o mecanismo de troca de mensagens.

- Interface *Agletcontext*: é usada pelo *Aglet* para recuperar informações sobre o equipamento e também para enviar mensagens, inclusive para outros *Aglets* ativos neste local. Corresponde ao conceito de agência, visto no Capítulo 3 deste trabalho.

A partir desta visão geral, pôde-se então, iniciar a construção da aplicação de gerenciamento com AM. Para tanto, foi utilizado como referência o documento (Lange, 1998a).

5.3.2. O desenvolvimento do Protótipo de AM

Depois da configuração do ambiente Java e do aplicativo Tahiti, com base na documentação, foram desenvolvidos exemplos de AM com o objetivo de testar as futuras necessidades de desenvolvimento dos protótipos. As funcionalidades implementadas e testadas foram:

a) os agentes são transferidos manualmente para um local de destino e lá mostram uma mensagem na tela;

b) os agentes recebem uma lista com itinerários a visitar, e em cada um deles é realizada a soma de um valor fixo a uma variável mostrada na tela;

c) os agentes visitam os elementos gerenciados, recolhem informações de variáveis locais e as armazenam em um vetor, que por sua vez é gravado em um arquivo texto na estação de gerenciamento ao final de sua execução.

Posteriormente, foram desenvolvidos protótipos que se constituíam de dois agentes, um estacionário (gerente) e outro móvel. Nesta ocasião, o gerente tinha a atribuição de iniciar o agente móvel, fornecendo a ele a relação de endereços a visitar. A troca de mensagens entre o agente móvel e o gerente também foi testada.

Em um terceiro momento passou-se a integrar as classes JDMK no protótipo do AM. Assim, o AM passou a enviar mensagens SNMP aos agentes SNMP ativos nos elementos gerenciados e a armazenar a resposta obtida em um vetor de respostas, que eram, ao final, enviadas ao gerente.

Após ter realizado a integração dos *Aglets* desenvolvidos e das classes do JDMK, a implementação das abordagens alternativas, que especificam o comportamento dos AM foi feita. Foram feitos então, testes com servidores de agentes, que receberiam as mensagens para iniciar um AM e ainda, o envio de dados ao gerente a partir de um conjunto de elementos visitados.

O comportamento dos protótipos desenvolvidos é comentado mais adiante, na Seção 6.2, durante a descrição dos experimentos realizados. Seu comportamento pode ser classificado como de operação reativa, além de autônomo em relação à sua movimentação, definições vistas nos conceitos de AM (Capítulo 3).

5.3.3. Dificuldades encontradas

A principal dificuldade encontrada durante o desenvolvimento do protótipo de AM foi a configuração da segurança, onde são definidas as classes que podem ser executadas por um AM.

Tal dificuldade foi devido à escassez de documentação. Este problema foi resolvido através da busca das informações junto à programadores da linguagem e em parte, pela adoção de um método de tentativas, até que se chegou a uma configuração ideal, que possibilitou a execução das classes Java nativas, *Aglets* e JDMK sem erros durante a execução de um AM.

A implementação também não conseguiu contemplar a apuração de tráfego recebido e enviado, pois todo o envio é realizado por um protocolo específico (ATP – *Agent Transmission Protocol*) cuja utilização não se conseguiu manipular via linhas de código em Java.

Assim, para contemplar a necessidade de apuração de tráfego gerado nos experimentos, optou-se por adotar a ferramenta de monitoração *CommView*¹³ 3.1. Tal ferramenta foi utilizada em paralelo à execução dos protótipos, com o objetivo de contabilizar os pacotes trocados em uma porta específica (no protótipo SNMP, 8085 e no protótipo de AM, 4434), o que permitiu que dados sobre o tráfego na estação de gerenciamento e nos elementos fossem conhecidos.

¹³ Produzido pela *TamoSoft Inc.* Disponível no endereço eletrônico: <http://www.tamos.com>

6. EXPERIMENTOS REALIZADOS

É evidente a necessidade de definir as vantagens advindas do uso de código móvel comparado a outras técnicas de gerência de redes (Fiorese, 2000). Vários trabalhos tentam validar o uso de AM na Gerência de Redes através de experimentos realizados de diversas formas.

A equipe chefiada pelo pesquisador Bernard Pagurek implementou uma infraestrutura para desenvolvimento de aplicações com código móvel e ainda um conjunto de ferramentas para a interação com agentes estacionários, localizados nos componentes gerenciados. Os trabalhos de Pagurek (2000), White (1998), Bieszczad (1998) e Susilo (1998) apresentam as experiências conseguidas com seus estudos e falam sobre o potencial uso de AM nas mais diversas atividades dentro do gerenciamento de redes. Tomarchio (2000) criou protótipos que utilizam AM para coletar informações sobre o estado da rede e para realizar o gerenciamento de recursos.

No artigo apresentado por Sahai (1998), a equipe de pesquisadores propõe uma alternativa de gerenciamento utilizando AM, em um modelo que visa principalmente a independência de localização da aplicação separando esta da estrutura e mecanismos de protocolo e das informações de gerenciamento. O principal objetivo deste modelo é permitir que o administrador possa visualizar dados de gerenciamento de qualquer equipamento, estando este na rede local ou conectado através de uma linha telefônica.

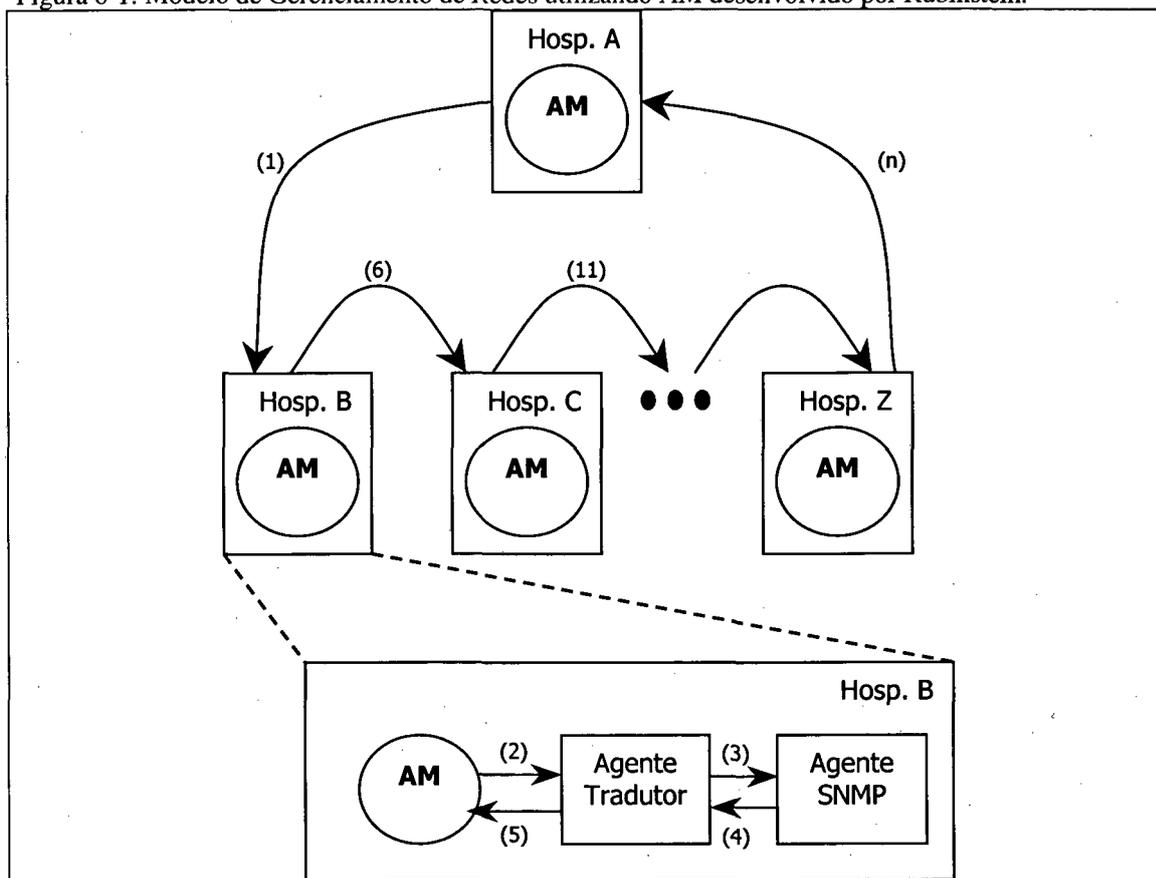
Em sua proposta, os AM são utilizados para prover autonomia das tarefas de gerenciamento, de forma que eles possam colher informações necessárias e executar interações na rede através de pedidos partidos da interface do administrador, mas iniciadas a partir do servidor. O desempenho dos AM é comparado com a gerência tradicional através da apuração da banda passante utilizada na rede e ainda, pela apuração do tempo de resposta na execução de uma tarefa qualquer. O comportamento dos AM é a mais simples possível: ele é iniciado no servidor, percorre todas as estações, e retorna ao servidor com as informações solicitadas.

Rubinstein realizou experimentos simulados sobre o uso de AM no gerenciamento de redes. Neles, também utilizou a abordagem mais simples (Rubinstein, 1999)

(Rubinstein, 1999a) e algumas estratégias para descarregamento (Rubinstein, 2000) (Rubinstein, 2000a), que indicam que o desempenho no tempo de resposta e na utilização de recursos da rede pode ser vantajoso em relação ao SNMP.

Em pesquisas mais recentes, Rubinstein (2000b, 2001) realizou implementações e as testou em uma rede. Para tanto, utilizou um modelo onde existe a figura de um agente tradutor, que traduzia as mensagens dos AM em mensagens SNMP, que eram então enviadas ao agente SNMP. Esta estrutura é representada na Figura 6-1.

Figura 6-1: Modelo de Gerenciamento de Redes utilizando AM desenvolvido por Rubinstein.



Fonte: (Rubinstein, 2001).

Em suas conclusões, esclareceu que o uso deste agente tradutor pode gerar um atraso adicional no tempo de resposta. Quanto ao protótipo SNMP implementado para fins de comparação, este realizava a troca de mensagens entre agentes e gerentes de forma síncrona, ou seja, o gerente aguardava até que a resposta fosse recebida para enviar a próxima requisição, procedimento diferente do disposto no modelo SNMP.

Fiorese (2000) apresenta experimentos realizados no laboratório Integrado de Informática do Centro Tecnológico (LICT) da UFSC onde são apurados os tempos totais na obtenção de variáveis. Neste trabalho, os experimentos não utilizam situações onde haja “fatores que interfiram no desempenho”, referindo-se a gargalos ou links lentos.

Adicionalmente aos trabalhos apresentados, os experimentos executados neste trabalho tiveram o objetivo de contribuir para a comprovação das vantagens do uso de AM para as atividades da gerência de redes.

Este trabalho realizou os experimentos em uma estrutura de rede real - a estrutura atual da Universidade Regional de Blumenau (FURB) – cujos aspectos de desempenho analisados sofreram interferências da latência de rede, carga dos nós, erros de transmissão e o tempo de processamento da camada de aplicação. Não obstante a isso, como diferencial, os experimentos utilizaram, além das situações de rede em que o AM pode apresentar maiores variações (segmentos com altas latências), a aplicação de estratégias no desenvolvimento do protótipo mencionadas na Seção 4.4.

Para fins de complementação do diferencial, este trabalho analisou também a utilização de recursos de rede (utilização de banda dos segmentos de gargalo e na rede como um todo) dos AM em comparação ao SNMP. Também diferentemente dos trabalhos anteriores, os protótipos fizeram a interação diretamente com os agentes SNMP estacionários, metodologia bastante aceita e referenciada nas leituras e propostas estudadas (Rubinstein, 1999) (Pagurek, 2000).

Para manter a compatibilidade com os experimentos anteriores (Costa, 1999), (Rubinstein, 2001) (Fiorese, 2000), todos os experimentos foram realizados dez (10) vezes com um número de elementos gerenciados que variou de um (1) a duzentos e cinquenta (250). Os valores analisados correspondem à média destes experimentos com cada conjunto de amostragem.

A tarefa de gerenciamento escolhida foi a recuperação da variável da MIB SNMP *ifInErrors* do ramo *Interfaces* da MIB-2. Devido à dificuldade de instalação e configuração dos aplicativos necessários aos testes em um número tão grande de elementos de rede, os experimentos utilizaram um conjunto de cinco elementos gerenciados, onde seus endereços foram colocados várias vezes no itinerário do AM com o propósito de fazê-lo interagir com

o número de elementos necessário ao experimento. O mesmo princípio também foi utilizado no protótipo SNMP.

É importante ressaltar que os mesmos endereços foram repetidos várias vezes no itinerário para que o AM tivesse o valor mais próximo ao real, porque por outro lado, se os endereços fossem repetidos via código, o tamanho inicial do AM não refletiria a realidade, alterando assim, os valores apurados de tráfego gerado e tráfego no enlace de gargalo.

O resumo do mapeamento dos experimentos realizados pode ser visto no Quadro 6-1. Este mapeamento é explicado em detalhes na seqüência.

Quadro 6-1: Mapeamento dos experimentos propostos.

Proposta	Gerência Local	Gerência Remota (com enlace de gargalo com menor banda passante)	Gerência Remota (com um enlace de gargalo com maior latência)
(A) SNMP	Tempo Resposta Tráfego Total	Tempo Resposta Tráfego Total Tráfego no gargalo/EG	Tempo Resposta Tráfego Total Tráfego no gargalo/EG
(B) AM retorna à Estação de Gerenciamento (EG) após tarefa	Tempo Resposta Tráfego Total	Tempo Resposta Tráfego Total Tráfego no gargalo/EG	Tempo Resposta Tráfego Total Tráfego no gargalo/EG
(C) AM envia dados à Estação de Gerenciamento após tarefa	Tempo Resposta Tráfego Total	Tempo Resposta Tráfego Total Tráfego no gargalo/EG	Tempo Resposta Tráfego Total Tráfego no gargalo/EG
(D) Usando uma estratégia de descarregamento: AM envia dados à estação de gerenciamento em um intervalo de nós visitados	Tempo Resposta Tráfego Total	Tempo Resposta Tráfego Total Tráfego no gargalo/EG	Tempo Resposta Tráfego Total Tráfego no gargalo/EG
(E) Uso de "servidor de agentes" para realizar tarefas em segmentos remotos	Não se aplica	Tempo Resposta Tráfego Total Tráfego no gargalo/EG	Tempo Resposta Tráfego Total Tráfego no gargalo/EG

6.1. Condições de Rede testadas

A partir do mapeamento dos experimentos, serão analisados as condições da rede testadas e também os motivos para que os experimentos fossem realizados em tal cenário. É importante ressaltar que pelo fato dos experimentos terem sido realizados em uma estrutura de rede real e ativa, a escolha das situações de rede se limitou às condições disponíveis.

6.1.1. Gerência Local

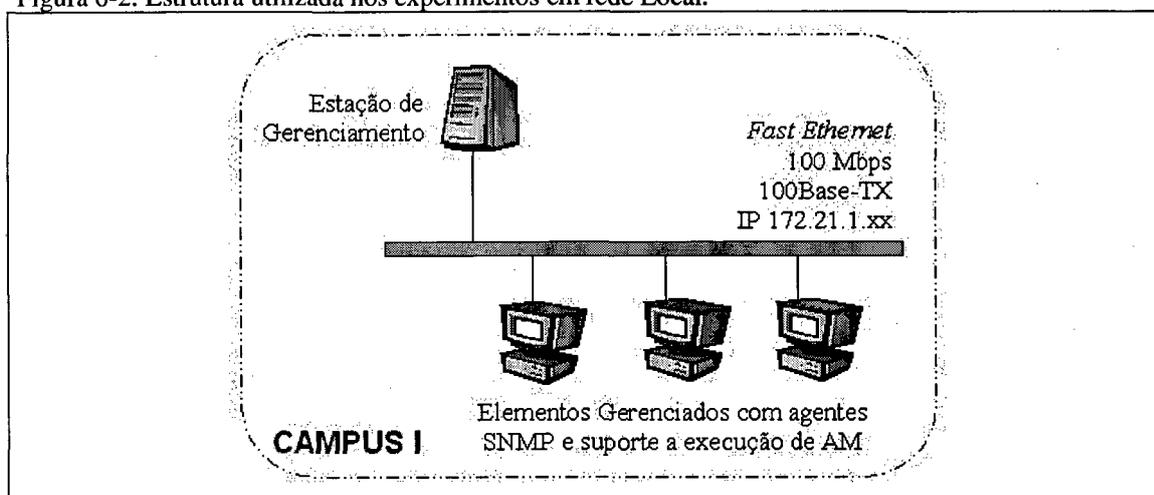
Mesmo sabendo que não se trata da condição de rede mais indicada para o uso de AM, com objetivo de comparação, a pesquisa acredita ser prudente verificar a diferença de comportamento em uma rede local (*Fast Ethernet*) dos dois protótipos: AM e SNMP. Este

experimento teve o objetivo de avaliar os protótipos em uma rede que apresenta a estação de gerenciamento e todos os elementos a serem gerenciados inseridos na rede local, onde a banda passante é homogênea (100 Mbps) e a latência de rede tende a zero (Costa, 1999).

Para este experimento, foi apurado além do tempo de resposta das operações, o tráfego na estação de gerenciamento, para quantificar a utilização de recursos da rede das propostas. Neste caso, a latência da rede foi apurada durante a realização dos experimentos e foi informada durante a explanação dos resultados obtidos com cada experimento.

Para estes experimentos foi utilizada a estrutura de rede atual do campus I da Universidade Regional de Blumenau. Os elementos gerenciados se encontram em vários setores e laboratórios localizados nas imediações do Núcleo de Informática, que mantém a estação de gerenciamento. Todos os elementos gerenciados e também a estação de gerenciamento estão no mesmo segmento de rede, como pode ser visto na Figura 6-2.

Figura 6-2: Estrutura utilizada nos experimentos em rede Local.



Todos os equipamentos utilizados são equipados com processadores padrão Intel, com velocidade variando de 300 a 700 Mhz e demais especificações conforme o Quadro 6-2. A estação de gerenciamento mantém uma conexão à rede local de 100 Mbps e a maioria dos elementos gerenciados estavam conectados a 100 Mbps.

Quadro 6-2: Especificação técnica dos equipamentos utilizados nos experimentos da gerência local.

Função	Especificações
Estação de Gerenciamento	Processador AMD K6-2 450 Mhz, Memória RAM 96 MB, Conexão 100 Mbps
Elemento Gerenciado 1	Processador Pentium III 600 Mhz, Memória RAM 256 MB, Conexão 100 Mbps
Elemento Gerenciado 2	Processador AMD Duron 750 Mhz, Memória RAM 256 MB, Conexão 10 Mbps
Elemento Gerenciado 3	Processador AMD Duron 750 Mhz, Memória RAM 128 MB, Conexão 10 Mbps
Elemento Gerenciado 4	Processador AMD Duron 750 Mhz, Memória RAM 256 MB, Conexão 100 Mbps
Elemento Gerenciado 5	Processador Intel Celeron 300 Mhz, Memória RAM 128 MB, Conexão 100 Mbps

6.1.2. Gerência Remota – Situação 1

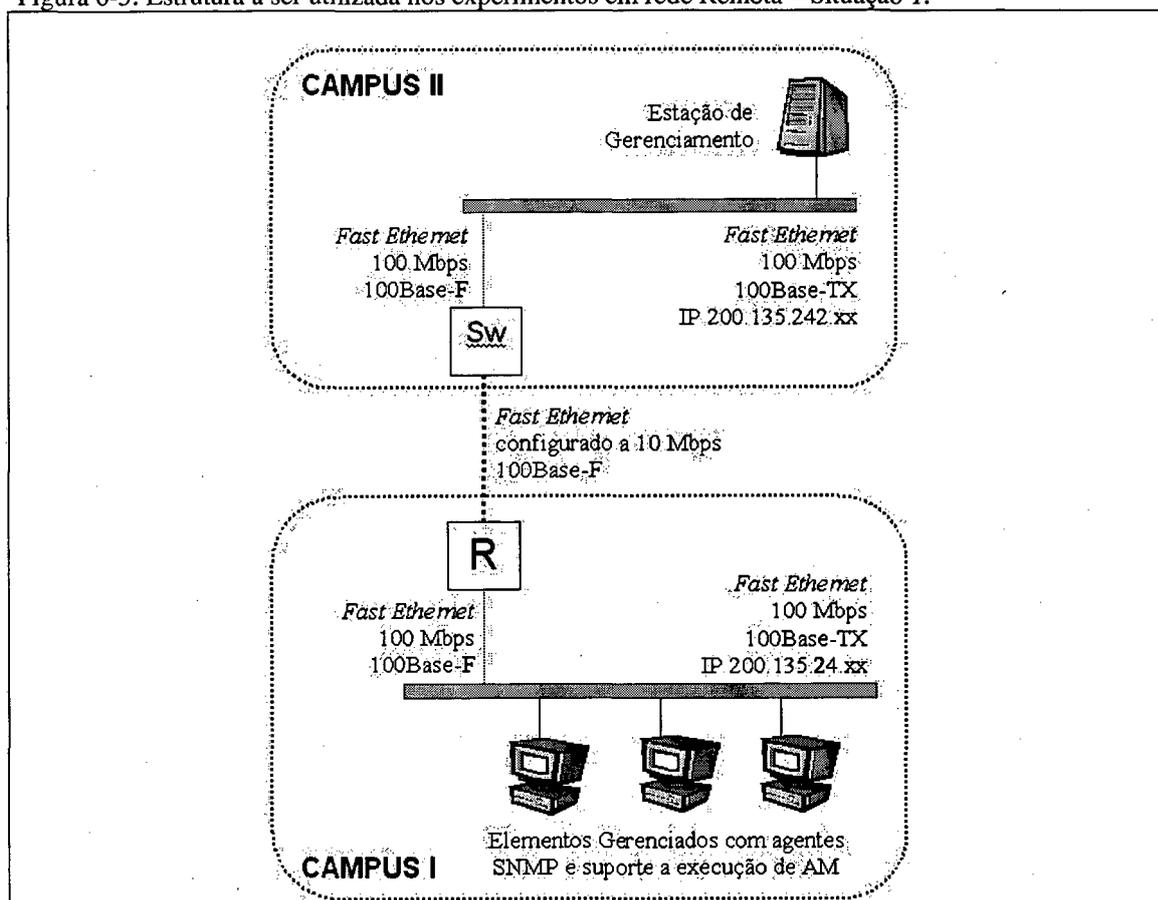
Uma condição de rede bastante comum é o gerenciamento remoto de uma rede, onde a estação de gerenciamento encontra-se ligada a um segmento (que por sua vez possui os elementos a serem gerenciados) através de uma conexão diferenciada, com maior latência e menor banda passante. Experimentos com este tipo de situação foram simulados por Costa (1999).

A intenção foi determinar se os protótipos (AM e SNMP) apresentam diferenças significativas em relação do tempo de resposta, quando existe um gargalo na rede a ser gerenciada.

Foi objetivo deste experimento também determinar se o protótipo de AM realmente representa economia de banda em relação ao SNMP no enlace de gargalo citado. Também nesta situação foi apurado o tráfego total gerado.

Para este experimento, foi usada a conexão existente na Universidade Regional de Blumenau entre o campus I e o campus II, ambos possuem redes locais *Ethernet* de 100 Mbps, mas são ligados por um link de 10 Mbps. A estação de gerenciamento localiza-se no segmento do campus II e as estações a serem gerenciadas, no campus I. A estrutura pode ser conhecida através da Figura 6-3.

Figura 6-3: Estrutura a ser utilizada nos experimentos em rede Remota – Situação 1.



A estação de gerenciamento foi o mesmo equipamento utilizado no experimento anterior, porém, os elementos gerenciados possuíam configuração diferenciada, conforme demonstra o Quadro 6-3.

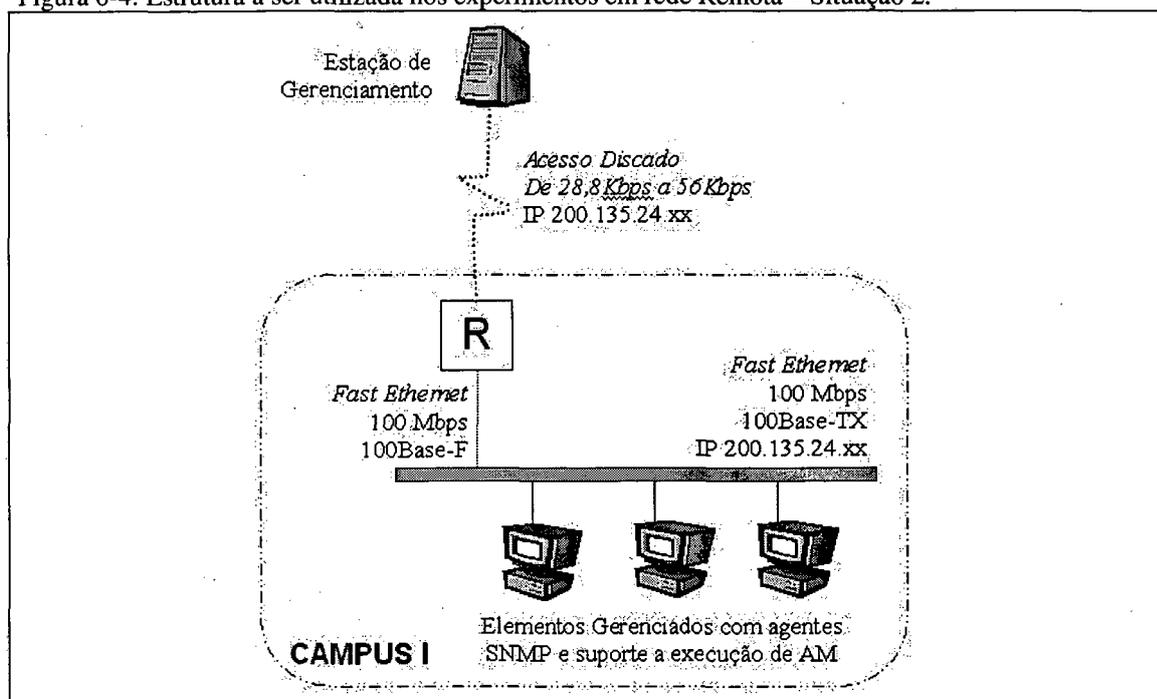
Quadro 6-3: Especificação técnica dos equipamentos utilizados nos experimentos da gerência remota.

Função	Especificações
Estação de Gerenciamento	Processador AMD K6-2 450 Mhz, Memória RAM 96 MB, Conexão 100 Mbps
Elementos Gerenciados (5 x)	Processador AMD K6-2 475 Mhz, Memória RAM 128 MB, Conexão 100 Mbps

6.1.3. Gerência Remota – Situação 2

A segunda situação em rede remota utilizada nos experimentos foi realizada ligando a estação de gerenciamento à rede onde encontravam-se os elementos gerenciados através de uma conexão via modem e acesso discado à rede, conforme pode ser visualizado na Figura 6-4.

Figura 6-4: Estrutura a ser utilizada nos experimentos em rede Remota – Situação 2.



Estes experimentos utilizaram exatamente os mesmos equipamentos da situação anterior. Tiveram como objetivo avaliar a diferença de comportamento em uma maior latência do que o experimento anterior. A largura de banda nesta conexão também foi drasticamente reduzida em relação ao experimento anterior, sendo registrada em 28,8 Kbps.

6.2. Abordagens de Implementação

Conforme mencionado na Seção 4.4, os protótipos de AM englobam as otimizações internas referente ao armazenamento de classes para o uso posterior (serviço provido pelo MAS do ambiente *Aglets*). Seu itinerário também é reduzido quando do envio das informações correspondentes à estação de gerenciamento.

Segue a descrição das abordagens de implementação que foram utilizadas nos experimentos.

6.2.1. SNMP tradicional (Protótipo A)

Para fins de comparação, o protótipo desenvolvido foi utilizado para recuperação de variáveis da MIB em um conjunto de elementos gerenciados. Seu funcionamento seguiu rigorosamente o modelo de gerenciamento SNMP, ou seja, a aplicação gerente envia uma

mensagem de solicitação a cada um dos elementos gerenciados da amostra e contabiliza o tempo necessário para a obtenção das variáveis distribuídas quando do recebimento da última mensagem de resposta enviada pelo agente SNMP.

6.2.2. AM retorna à Estação de Gerenciamento após tarefa (Protótipo B)

Nesta abordagem, a aplicação de gerenciamento solicita ao AM que o mesmo recupere a variável da MIB SNMP de uma lista de elementos gerenciados. A partir de então, o AM se transfere para o primeiro elemento, e neste local, enviará uma solicitação SNMP para o agente SNMP que retornará a ele o valor correspondente.

O valor obtido será então armazenado pelo AM em um vetor, e será transferido junto com o mesmo até o último elemento gerenciado a ser visitado, quando então, o AM retornará a estação de gerenciamento com todos os dados colhidos.

A realização deste experimento surge da necessidade de apurar dados comparativos da abordagem tradicional de implementação de AM para prosseguir com a comparação das estratégias de implementação seguintes.

6.2.3. AM envia dados à Estação de Gerenciamento após tarefa (Protótipo C)

Esta abordagem teve basicamente o mesmo comportamento da anterior, porém, ao final da coleta de informações, o AM simplesmente envia os dados colhidos à estação de gerenciamento e finaliza sua execução estando ainda no último elemento visitado.

O objetivo deste experimento é identificar se o envio somente dos dados à estação de gerenciamento significa uma melhoria de desempenho (tempo e tráfego gerado) em relação ao SNMP, já que é evidente sua vantagem em relação à implementação tradicional de AM.

6.2.4. AM envia dados à Estação de Gerenciamento em um intervalo de nós visitados (Protótipo D)

Nesta abordagem, a partir da saída do AM da estação de gerenciamento, em um intervalo fixo de nós ele envia os dados colhidos até aquele momento à EG, de forma que a quantidade de dados colhidos não se acumule, o que pode representar uma melhora

significativa de desempenho na utilização de AM, pois tornaria a transferência do AM mais rápida. O intervalo de envio de dados utilizados nos experimentos foi de 20% do itinerário do AM, mas outros intervalos tiveram uma análise teórica.

Esta abordagem foi comparada com o experimento anterior e ainda com o modelo SNMP tradicional.

6.2.5. Utilização de um “servidor de AM” (Protótipo E)

Este experimento somente foi realizado com a gerência remota, pois sua utilização se justifica no caso de haver um segmento mais lento separando a estação de gerenciamento dos elementos gerenciados.

O comportamento do modelo de AM é ligeiramente diferente: a estação de gerenciamento envia uma solicitação a um servidor de agentes, que se encontra do outro lado do segmento “problemático”. A partir disso, este servidor inicia a execução de um AM que executa seu itinerário e visita todos os elementos gerenciados, colhendo as informações solicitadas. Após o término da tarefa, o AM envia os dados à estação de gerenciamento (através do enlace de gargalo) e finaliza sua execução.

O objetivo deste experimento é determinar quanto o uso desta abordagem pode ser vantajoso no sentido de economizar um segmento ou gargalo específico, em relação a todas as abordagens anteriores.

6.3. Variáveis Apuradas

As variáveis consideradas para a comparação de desempenho, apuradas no decorrer dos experimentos são descritas na seqüência.

6.3.1. Tempo de Resposta

O tempo de resposta no experimento SNMP foi obtido a partir do envio da primeira mensagem SNMP ao primeiro nó da rede a que se refere à tarefa e foi computado quando do recebimento da última mensagem SNMP recebida pelo gerente. Já o tempo de resposta de uma operação de gerenciamento que utiliza AM foi computado a partir da solicitação inicial feita pelo gerente até o recebimento da resposta por este.

Dentro deste período, podem-se enumerar os eventos que seguem:

- a) a serialização¹⁴ do agente e o seu envio à primeira agência de destino ou ainda, o envio da mensagem de solicitação da tarefa pelo gerente à agência onde está alojado o AM;
- b) a desserialização¹⁵ do AM no destino e a restauração de seu estado;
- c) a execução da tarefa, ou seja, o envio da solicitação ao agente SNMP local, que requisita a variável da MIB e seu recebimento e armazenamento;
- d) a serialização do AM na agência e seu envio à próxima agência;
- e) a repetição dos passos B, C e D até que todos os nós a que se referem à tarefa sejam visitados;
- f) o envio à aplicação gerente dos dados colhidos durante todo o processo ou o retorno do AM propriamente dito;
- g) o recebimento dos dados ou do AM pela aplicação gerente.

O objetivo na apuração desta variável é obviamente determinar em quais situações ou condições os AM apresentam melhor desempenho em relação ao tempo.

6.3.2. Tráfego Total Gerado

O tráfego total gerado durante o experimento com SNMP será obtido a partir da contabilização da soma dos pacotes SNMP recebidos e enviados pela estação de gerenciamento. Já no experimento que utiliza AM o valor desta variável será apurado somando-se aos pacotes recebidos e enviados pela EG, o valor enviado por cada um dos elementos gerenciados.

O objetivo da apuração desta variável é verificar qual a taxa de utilização dos recursos de rede na utilização de AM em operações de gerenciamento em relação ao SNMP.

¹⁴ Serialização é o armazenamento das informações do processo de execução de um agente, de forma que o mesmo possa ser reconstruído em outro meio.

¹⁵ Desserialização é o processo inverso da serialização, ou seja, a partir das informações armazenadas, reconstruir o processo de execução de um agente.

6.3.3. Tráfego no Enlace de Gargalo

O objetivo na apuração desta variável foi avaliar se o uso de AM realmente pode poupar o uso de um segmento mais lento, que geram normalmente problemas no modelo SNMP tradicional, ou seja, se a nova proposta é realmente indicada para esta situação de rede (gerência remota).

Para o experimento utilizando o modelo SNMP tradicional, o tráfego no enlace de gargalo será o tráfego total gerado, ou seja, a soma do tamanho dos pacotes emitidos e recebidos pela estação de gerenciamento, que chegarão até os elementos gerenciados através deste segmento lento.

Já com relação ao modelo que utiliza AM, o tráfego no enlace de gargalo será obtido a partir da soma do tamanho da mensagem de solicitação ou do AM enviado pela estação de gerenciamento ao primeiro elemento gerenciado do outro lado do segmento, e das respostas enviadas pelo AM à estação de gerenciamento após a execução de sua tarefa.

7. RESULTADOS OBTIDOS E ANÁLISE DOS RESULTADOS

Uma dificuldade encontrada durante os experimentos foi a carga da rede, que por vezes gerou resultados incomparáveis entre os protótipos. Para que fosse possível agrupar dados que pudessem ser comparados, os experimentos aconteceram sempre no mesmo período do dia.

É importante lembrar que o protótipo SNMP desenvolvido na linguagem Java não é o padrão utilizado pelos softwares hoje existentes no mercado, que implementam o SNMP com tempos de resposta melhores. Sendo Java uma linguagem interpretada, seu tempo de resposta na execução é ligeiramente maior do que o de linguagens compiladas. Mesmo assim, optou-se por desenvolver em Java para que os protótipos tivessem seu funcionamento e testes em igualdade de condições.

Poder-se-ia por outro lado, utilizar aplicativos SNMP disponíveis, porém, os tempos de resposta do AM seriam acrescidos do tempo de interpretação da linguagem, o que provavelmente geraria distorções na avaliação.

É importante lembrar que toda a análise das variáveis de desempenho e as conclusões obtidas a partir dela são oriundas especificamente dos protótipos desenvolvidos, e que, ao implementar protótipos em plataformas diferentes, todas as variáveis podem ter comportamentos diferentes.

A apresentação dos resultados obtidos e a análise dos resultados foram montadas de acordo com cada variável de desempenho.

7.1. Análise do Tráfego Total Gerado

Um dos fatores importantes a serem comparados é a utilização de recursos da rede independentemente da situação de rede testada, pois existe a preocupação em verificar se o uso de AM causará maior sobrecarga na rede como um todo, situação não desejada que pode agravar os problemas que a gerência de redes se propõe a resolver.

Como já mencionado anteriormente, o tamanho inicial do AM é geralmente muito superior ao tamanho das mensagens SNMP. Assim, mesmo tendo ciência de que o SNMP

tem uma grande vantagem neste quesito em relação ao AM, o objetivo de se apurar estes valores é avaliar a diferença de utilização dos recursos de rede das duas alternativas, fator que deve ser levado em consideração quando da escolha da abordagem para a realização de alguma tarefa de gerenciamento.

Os valores apurados no decorrer dos três experimentos são apresentados no Quadro 7-1. Com relação aos protótipos de AM, estes valores foram obtidos através da soma de bytes enviados por todos os equipamentos envolvidos nos experimentos. Já no protótipo SNMP, o valor total de tráfego refere-se aos bytes recebidos e enviados pela estação de gerenciamento.

Quadro 7-1: Valores em bytes de tráfego total obtidos nos experimentos.

Nós	SNMP	AM-B	AM-C	AM-D	AM-E
2	164	20.900	11.738	5.326	5.580
6	492	58.470	29.692	16.080	16.247
10	822	82.954	61.877	35.226	35.405
20	1.644	174.608	137.541	73.726	73.935
40	3.288	402.142	333.967	173.485	173.754
60	4.932	689.241	593.090	305.543	305.872
80	6.576	976.340	912.418	466.214	466.603
100	8.220	1.454.010	1.299.527	673.209	673.658
120	9.864	1.804.633	1.733.767	895.114	895.623
140	11.508	2.487.017	2.232.397	1.147.488	1.148.057
160	13.267	3.169.402	2.793.567	1.431.620	1.432.249
180	14.936	3.851.786	3.463.207	1.778.837	1.779.526
200	16.587	4.451.762	4.145.927	2.140.386	2.141.135
220	18.264	5.334.926	4.872.407	2.520.926	2.521.735
250	20.550	6.478.324	6.088.727	3.141.688	3.142.587

O tamanho de cada mensagem SNMP apurada foi de 41 bytes, sendo assim, para a obtenção da variável SNMP, são trocadas entre a estação de gerenciamento e elemento gerenciado as mensagens de solicitação (S) e de resposta (R), o que totaliza em média 82 bytes. Pode-se deduzir então, a relação matemática para o protótipo SNMP desenvolvido (representada na Equação 1), onde n representa o número de elementos gerenciados envolvidos na tarefa de gerenciamento.

$$\begin{aligned}
 \text{TrafegoSNMP} &= (S + R) * n \\
 &= (41 + 41) * n \\
 &= 82n
 \end{aligned}
 \tag{Equação 1}$$

Assim, o tráfego total gerado pelo SNMP (protótipo A) será dado sempre através da multiplicação entre o número de elementos gerenciados e a soma do tamanho das mensagens de solicitação e resposta, no caso 82 bytes.

Os tamanhos iniciais dos AM apurados foram de 6.818, 5.352, 5.288 e 5.513 bytes, respectivamente para os protótipos B, C, D e E. Considerando que este montante deverá ser transferido para cada elemento da rede a ser gerenciado, pode-se verificar que o tráfego total gerado será sempre superior ao do SNMP, seja para qualquer número de elementos a serem gerenciados, mesmo sem considerar o tamanho do itinerário e das respostas obtidas, informações que são carregadas juntamente com o AM.

Os gráficos apresentados na Figura 7-1 e na Figura 7-2 mostram a comparação dos valores de tráfego total gerado pelos protótipos durante os experimentos, onde é possível perceber que os valores obtidos com o SNMP são muito inferiores aos apurados nos experimentos com AM.

Figura 7-1: Comparação entre os valores de tráfego total nos experimentos do SNMP (protótipo A) e do AM tradicional (protótipo B).

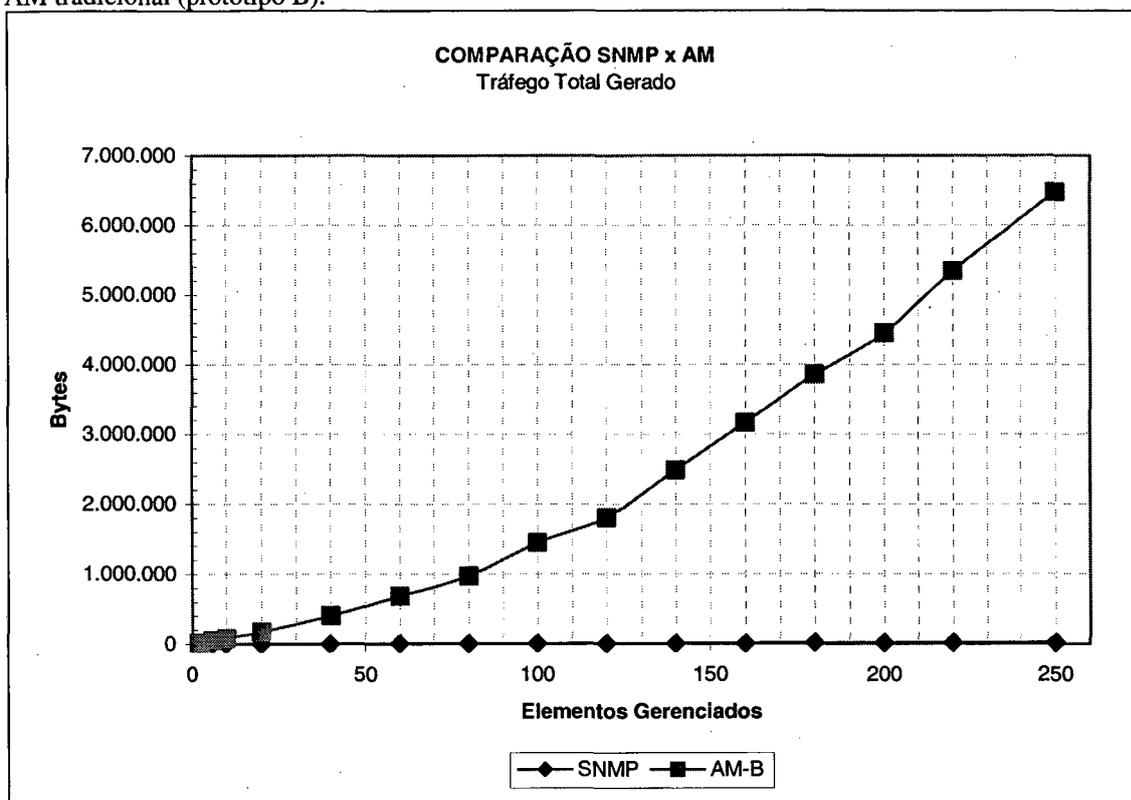
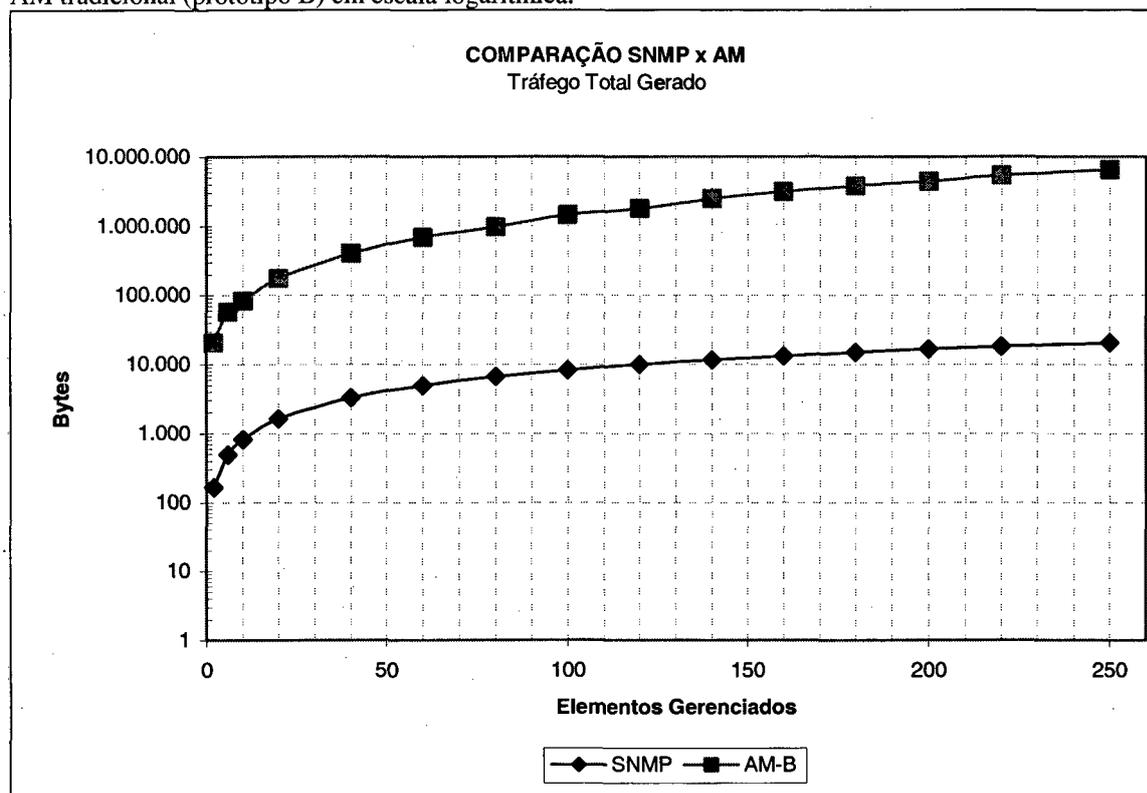


Figura 7-2: Comparação entre os valores de tráfego total nos experimentos do SNMP (protótipo A) e do AM tradicional (protótipo B) em escala logarítmica.



O tráfego gerado pelo AM tradicional é, em média, 187 vezes maior do que o tráfego gerado pelo SNMP. Referindo-se a percentuais, o tráfego gerado pelo protótipo SNMP representa em média 0,6147% da quantidade de tráfego gerada pelo AM tradicional. Ou seja, a cada 10 bytes que o protótipo SNMP transfere através da rede, o protótipo do AM tradicional transfere 1.626 bytes.

Ainda com relação ao tráfego total, é possível perceber a considerável diminuição no consumo de recursos da rede em decorrência das otimizações realizadas na implementação do AM através da Figura 7-3 e da Figura 7-4.

Figura 7-3: Comparação entre os valores de tráfego total dos experimentos com diferentes abordagens de implementação.

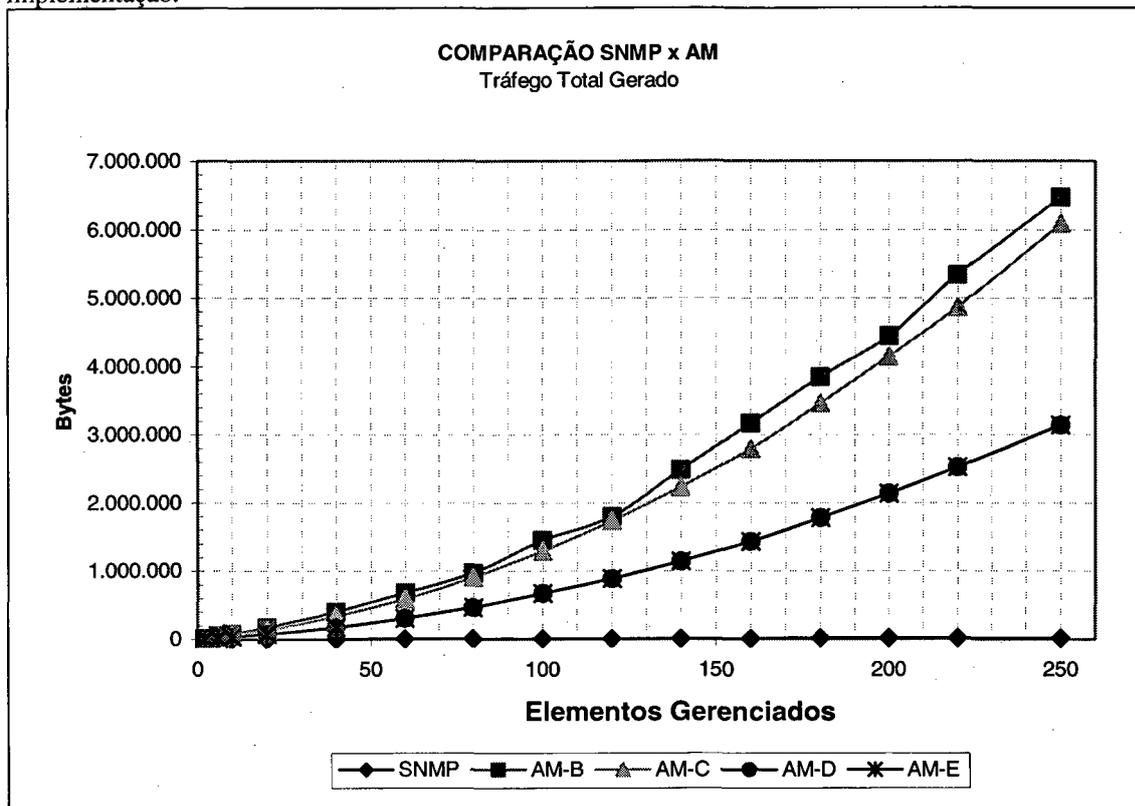
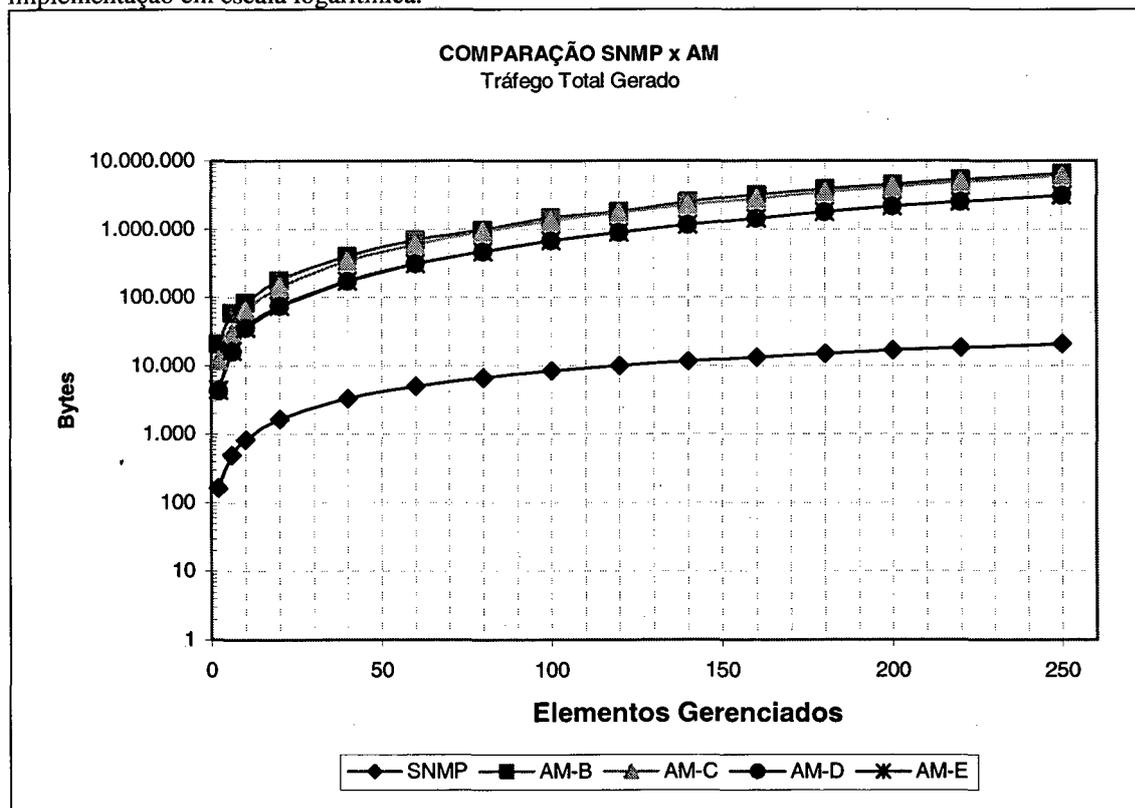


Figura 7-4: Comparação entre os valores de tráfego total dos experimentos com diferentes abordagens de implementação em escala logarítmica.



Para o protótipo C (onde o AM envia os resultados à estação de gerenciamento ao final da tarefa), a redução do tráfego em relação ao SNMP tradicional (protótipo B) deveria obviamente representar o tamanho do AM que não retorna à EG. Porém, o fato dos resultados finais não retornarem à estação de gerenciamento como uma variável interna do AM implica no envio das mesmas através de uma mensagem direcionada à estação de gerenciamento, o que necessita da adição de um cabeçalho, fazendo com que esta diferença de tráfego diminua.

Em um exemplo ilustrativo, no caso do AM tradicional (protótipo B), a estação de gerenciamento recebe ao final da tarefa 8K – que representa 5K do AM adicionados a 3K das respostas –, no protótipo C a estação de gerenciamento recebe além das respostas (3K), um cabeçalho da mensagem (1K), diminuindo o tráfego apurado dos protótipos B e C de 8K para 4K, e não para 3K como se poderia inicialmente imaginar. A redução do tráfego entre os protótipos B e C é, em média, de 16,3%.

É pertinente avaliar a considerável diminuição do tráfego total gerado pela adoção de uma estratégia de descarregamento como a adotada nos protótipos D e E. Neles, a partir da visita de um número estipulado de elementos do itinerário, os dados são enviados à EG, e conseqüentemente deixam de ser transferidos junto com o AM, facilitando inclusive sua migração através dos elementos para a continuidade da execução da tarefa. Nestes protótipos também foi adotado o método de envio dos dados colhidos à EG, sem a necessidade do código retornar ao final da tarefa.

O uso destas abordagens conseguiu reduzir, em média, 57,00% a utilização de recursos em comparação ao AM tradicional (protótipo B) e em 48,80% em relação ao protótipo C, mas ainda é muito superior ao SNMP (protótipo A). Seu tráfego é, em média, 83 vezes maior em relação ao SNMP, cujo tráfego representa somente 1,58% do tráfego gerado pelos protótipos D e E. Assim, para cada 10 bytes transmitidos pelo protótipo SNMP, os protótipos D e E transferem, em média, 632 bytes.

A diferença de tráfego entre os protótipos D e E é dada pela adição do tamanho da solicitação inicial usada no protótipo E (apurada através do software *CommView* em 149 bytes), somada ao vetor de itinerário, que é enviado pela estação de gerenciamento até o servidor de agentes de onde é iniciada a tarefa.

Também é importante lembrar que todas as implementações de AM realizaram a revisão do vetor de itinerário quando do envio dos dados à estação de gerenciamento no sentido de eliminar os endereços já visitados, o que contribuiu para a redução do tráfego.

7.2. Análise do Tráfego na Estação de Gerenciamento/Enlace de Gargalo

Uma importante vantagem atribuída aos AM é sua flexibilidade de implementação, que permite conseguir um melhor desempenho em determinadas condições desfavoráveis de rede, como na existência de gargalos, com menor banda passante ou maior latência. Este melhor desempenho estaria associado ao menor uso do segmento “problemático”, caracterizado pela passagem do AM uma única vez no segmento ou ainda, por outros métodos de implementação já descritos neste trabalho (Seção 4.4).

Da mesma forma, os AM teriam condições de diminuir o tráfego na estação de gerenciamento, que freqüentemente tem problemas de sobrecarga no modelo de gerência SNMP, utilizando-se dos mesmos artifícios.

Esta variável de desempenho foi apurada considerando o somatório dos bytes recebidos e enviados pela estação de gerenciamento, o qual – particularmente nas situações testadas por este trabalho com relação à gerência remota – é o mesmo valor transmitido pelo enlace de gargalo. Os valores apurados podem ser vistos no Quadro 7-2.

Quadro 7-2: Valores em bytes de tráfego na estação de gerenciamento ou enlace de gargalo apurados nos experimentos.

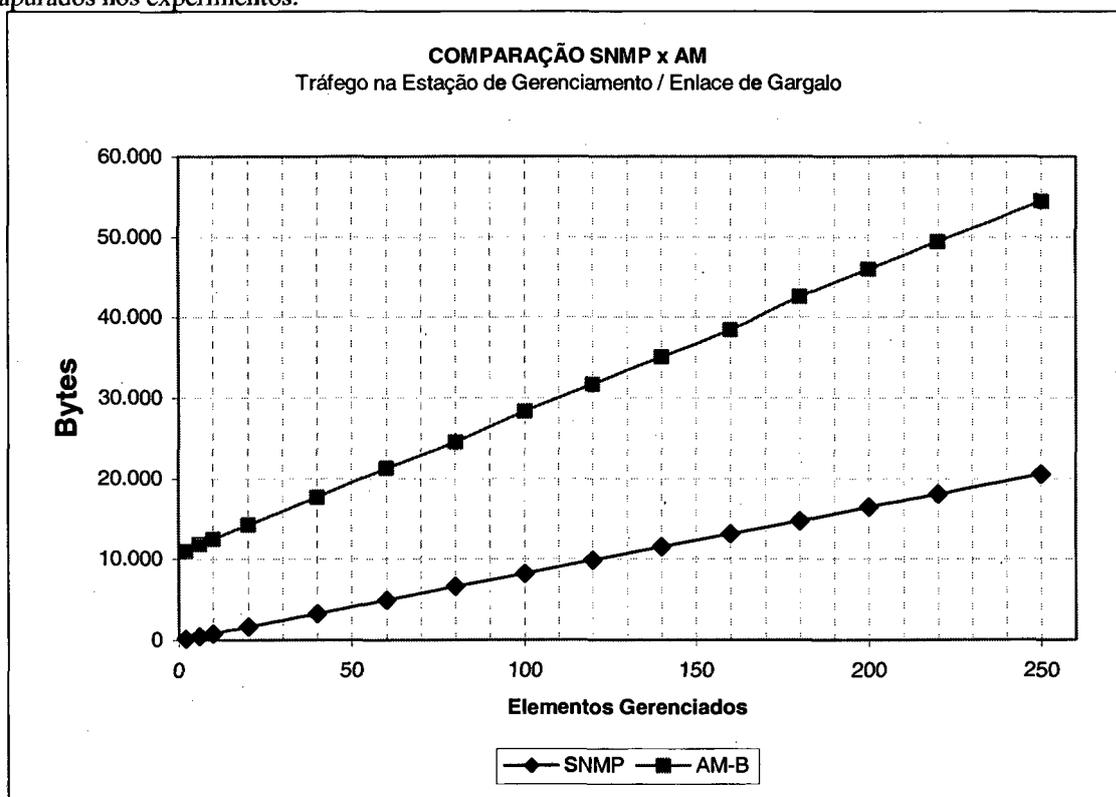
TRÁFEGO NA ESTAÇÃO DE GERENCIAMENTO / ENLACE DE GARGALO					
NOS	SNMP	AM-B	AM-C	AM-D	AM-E
2	164	10.948	6.246	6.898	3.940
6	492	11.836	6.568	7.188	4.174
10	822	12.484	6.890	7.480	4.564
20	1.644	14.223	7.777	8.286	5.344
40	3.288	17.674	9.473	9.786	6.904
60	4.932	21.227	11.217	11.434	8.464
80	6.576	24.531	12.881	13.402	10.024
100	8.220	28.308	14.760	15.647	11.584
120	9.864	31.710	16.478	17.591	13.144
140	11.508	35.059	18.097	19.698	14.704
160	13.152	38.466	19.844	21.696	16.264
180	14.736	42.556	21.844	24.013	17.824
200	16.440	46.016	23.562	26.014	19.384
220	18.084	49.420	25.325	28.066	20.944
250	20.550	54.460	27.812	31.091	23.284

Com relação ao SNMP (protótipo A) e ao AM tradicional (protótipo B) quanto ao tráfego na estação de gerenciamento, pode-se verificar que a diferença entre os valores apurados aumenta conforme cresce o número de elementos a serem gerenciados, como também pode ser visto na Figura 7-5.

A diferença entre o tráfego gerado pelo SNMP e pelo AM tradicional é calculada em 75,71% em média, dentre a quantidade de elementos gerenciados analisada, com vantagem para o protótipo SNMP. Diante deste comportamento, pode-se realizar uma análise teórica das expressões que refletem o comportamento do tráfego gerado na estação de gerenciamento ou enlace de gargalo.

Igualmente ao tráfego total para o protótipo SNMP já analisado na Seção 7.1, o tráfego na estação de gerenciamento ou enlace de gargalo para o protótipo SNMP é dado pela Equação 2.

Figura 7-5: Comparação entre valores de tráfego na estação de gerenciamento ou enlace de gargalo apurados nos experimentos.



$$\begin{aligned}
 \text{TráfegoEG}_{SNMP} &= (S + R) * n \\
 &= (41 + 41) * n \\
 &= 82n
 \end{aligned}$$

Equação 2

Ou seja, para o protótipo SNMP construído, o tráfego na estação de gerenciamento pode ser calculado pela soma dos tamanhos das mensagens de solicitação (**S**) e de resposta (**R**), multiplicada pela quantidade de elementos a serem gerenciados (**n**).

Com relação ao protótipo do AM tradicional (protótipo B), pode-se verificar que o tráfego na estação de gerenciamento é calculado pela relação dos elementos listados no Quadro 7-3. A relação entre estes elementos pode ser conhecida na Equação 3.

Quadro 7-3: Relação dos elementos que definem o tráfego na estação de gerenciamento para o protótipo B e seus valores apurados durante os experimentos.

Fator	Descrição	Valor Apurado nos experimentos com o software CommView
AM	Tamanho do código do AM do protótipo B	6818 bytes
I	Tamanho de um elemento de itinerário	75 bytes
R	Tamanho de uma resposta obtida	74 bytes
n	Número de elementos gerenciados	-

$$\begin{aligned}
 \text{TrafegoEG}_B &= AM + (I * n) + AM + (R * n) \\
 &= 6818 + (75 * n) + 6818 + (74 * n) \\
 &= 149n + 13636
 \end{aligned}
 \tag{Equação 3}$$

Ou seja, o tráfego na estação de gerenciamento ou enlace de gargalo para o protótipo B é calculado em dois estágios, onde: o primeiro (ida) é a soma do tamanho do AM com o vetor de itinerários e o segundo (volta) é dado pela soma do tamanho do AM com as respostas.

Na análise das equações anteriores (Equação 2 e Equação 3), pode-se verificar através da Equação 4, que o tráfego na estação de gerenciamento gerado pelo SNMP (protótipo A), jamais será maior do que o tráfego gerado pelo AM tradicional (protótipo B), pois qualquer que seja o número de elementos gerenciados, o resultado da equação do AM tradicional será maior. Portanto, a quantidade de bytes transitando pela estação de gerenciamento ou segmento de gargalo será maior.

$$\begin{aligned}
 \text{TrafegoEG}_{SNMP} &< \text{TrafegoEG}_B \\
 82n &< 149n + 13636
 \end{aligned}
 \tag{Equação 4}$$

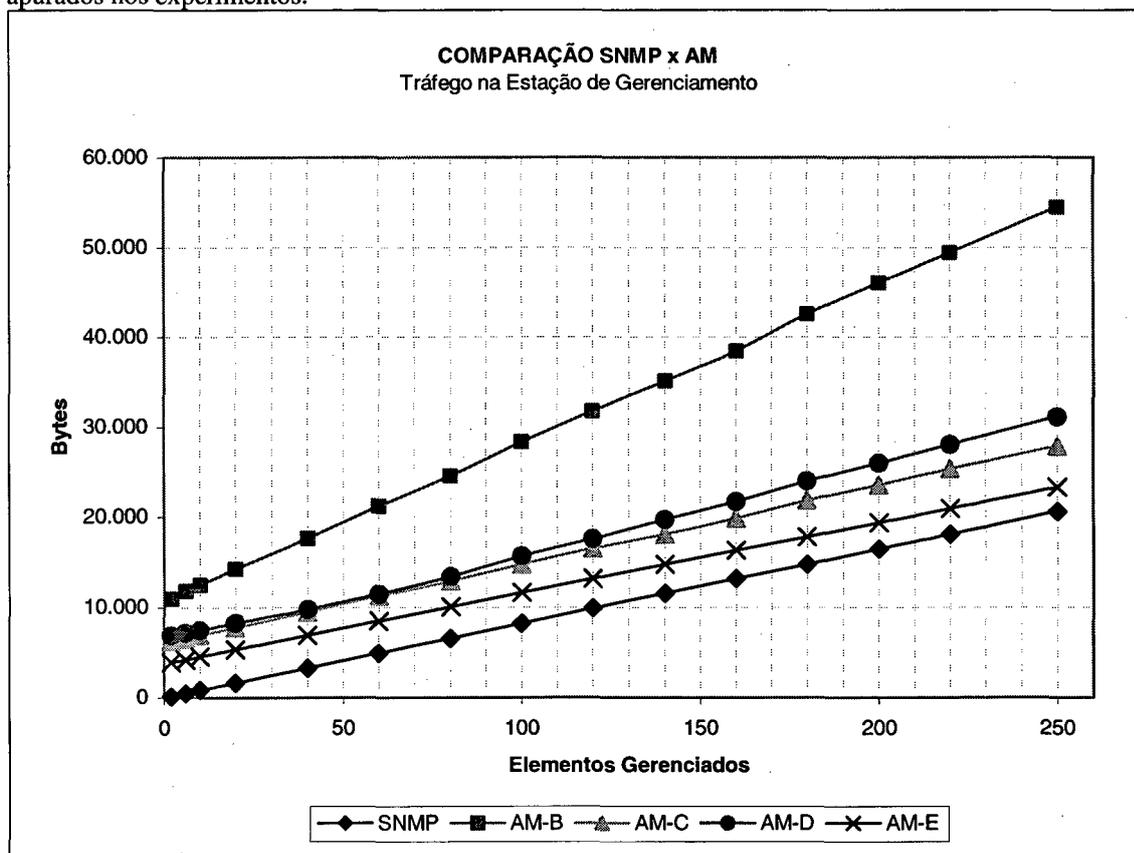
Com relação às outras abordagens de implementação, o resultado dos experimentos pode ser visto na Figura 7-6.

Pode-se perceber que os protótipos C e D – apesar de possuírem valores de tráfego maiores do que o SNMP no intervalo de elementos utilizado nos experimentos – possuem uma diferença decrescente, o que indica que existe uma quantidade de elementos gerenciados em que o tráfego SNMP passará a ser maior do que o tráfego dos protótipos C e D.

Através da análise teórica das equações, pode-se determinar a quantidade de elementos com a qual o protótipo SNMP ultrapassa os valores de tráfego obtidos com os protótipos C e D.

Inicialmente, para o protótipo C desenvolvido, o tráfego na estação de gerenciamento ou enlace de gargalo pode ser conhecido através da Equação 5. A principal diferença com relação ao protótipo anterior (B) é que o retorno dos resultados é feito através de uma mensagem (m), que possui um cabeçalho com tamanho apurado em 727 bytes, no lugar do retorno do código do AM.

Figura 7-6: Comparação entre valores de tráfego na estação de gerenciamento ou enlace de gargalo apurados nos experimentos.



$$\begin{aligned}
 \text{TrafegoEG}_C &= AM + (I * n) + m + (R * n) \\
 &= 5352 + (75 * n) + 727 + (3 * n) \\
 &= 78n + 6079
 \end{aligned}$$

Equação 5

Com relação ao tamanho das respostas, nota-se que para esta forma de envio (através de uma mensagem) o tamanho apurado de cada resposta é de 3 bytes, diferente do protótipo anterior, onde as respostas eram armazenadas em uma variável do AM e tinham um tamanho de 74 bytes.

Desta forma, a partir das equações referentes aos protótipos SNMP e AM-C, a determinação da quantidade de elementos com a qual o protótipo SNMP ultrapassa o valor de tráfego obtido com o protótipo C pode ser conhecida através da Equação 6 .

$$\begin{aligned}
 82n &= 78n + 6079 \\
 82n - 78n &= 6079 \\
 n &= \frac{6079}{4} \\
 n &= 1519
 \end{aligned}$$

Equação 6

Assim, com um número de 1519 elementos gerenciados, o tráfego na estação de gerenciamento gerado pelo protótipo C se iguala ao tráfego gerado pelo protótipo do SNMP. Acima de 1519 elementos, o protótipo C apresenta valores de tráfego na estação de gerenciamento inferiores aos apresentados pelo protótipo SNMP.

Dentro do número de elementos gerenciados utilizado nos experimentos, o uso do protótipo C conseguiu reduzir para 53,28% a margem de diferença em relação ao SNMP (protótipo A).

Da mesma forma, pode-se fazer a mesma relação com o protótipo D, onde o tráfego gerado na estação de gerenciamento é dado pela Equação 7. Nesta equação, a quantidade de respostas acumuladas antes do envio à estação de gerenciamento é estabelecida através de um percentual prédefinido do itinerário, inicialmente configurado em 20%, o que faz com que 5 mensagens com respostas sejam enviadas a estação de gerenciamento durante todo o período de execução da tarefa. O tamanho do AM para este protótipo é de 5288 bytes, enquanto que o tamanho do itinerário, do cabeçalho das mensagens e das respostas não variou.

$$\begin{aligned}
 \text{TrafegoEG}_D &= AM + (I * n) + \{[n/(n*0,2)] * m\} + (R * n) \\
 &= 5288 + (75 * n) + \{[n/(n*0,2)] * 727\} + (3 * n) \\
 &= 78n + 8923
 \end{aligned}$$

Equação 7

Para que o protótipo acumule 30% do itinerário antes de enviar as respostas para a EG, a equação modifica-se, porque o número de mensagens enviadas à estação de gerenciamento com resultados cai para 4. O mesmo raciocínio é tomado quando configurarmos o protótipo para enviar as respostas à estação de gerenciamento a cada 10% do itinerário cumprido, quando então, o número de mensagens à estação de gerenciamento será 10. Este valor deverá ser multiplicado ao tamanho do cabeçalho das mensagens para encontrar o tráfego que a estação de gerenciamento irá receber.

Assim, a partir das equações referentes aos protótipos SNMP e AM-D, a determinação da quantidade de elementos com a qual o protótipo SNMP ultrapassa o valor de tráfego obtido com o protótipo D (reportando a cada 20% do itinerário apurado) pode ser conhecida através da Equação 8.

$$\begin{aligned}
 82n &= 78n + 8923 \\
 82n - 78n &= 8923 \\
 n &= \frac{8923}{4} \\
 n &= 2231
 \end{aligned}$$

Equação 8

Desta maneira, utilizando o protótipo D e fazendo com que o mesmo reporte os resultados apurados à estação de gerenciamento cada vez que completar uma quantidade de elementos equivalente a 20% de seu itinerário, o mesmo terá um desempenho melhor em relação ao tráfego gerado na estação de gerenciamento do que o protótipo SNMP com um número de elementos gerenciados acima de 2231. Calculando projeções no mesmo sentido, com o AM reportando resultados a cada 30% de seu itinerário, seriam necessários no mínimo 1928 elementos para que o AM (protótipo D) tenha um tráfego menor do que o protótipo SNMP. Para 10% do itinerário seriam necessários 3140 elementos, enquanto que para 5% do itinerário seriam necessários 4957 elementos.

As projeções apontam que, se o objetivo principal é poupar a estação de gerenciamento, o AM (protótipo D) deve reportar os resultados em um intervalo maior de elementos visitados, porém, isso pode prejudicar o tempo de resposta, visto que a migração de um AM muito grande é demorada.

Quanto ao último protótipo de AM (protótipo E), o gráfico da Figura 7-6 mostrou que os valores de tráfego apurados na estação de gerenciamento ou no enlace de gargalo são realmente bem menores e têm seu crescimento mais modesto, mesmo quando o número de elementos gerenciados aumenta. A margem de diferença com relação ao protótipo SNMP cai para 41%, no intervalo de elementos gerenciados utilizados nos experimentos.

No caso específico do comportamento desta implementação de AM (protótipo E), o uso do enlace de gargalo seria o menor, porque nele trafegam somente a solicitação inicial com o vetor de itinerário enviado pela estação de gerenciamento e retornariam as mensagens com os resultados obtidos. Para este protótipo também foi adotado o comportamento de reportagem de resultados a cada 20% do itinerário visitado.

Desta forma, a equação do tráfego do protótipo E é obtida pela soma da solicitação inicial (S), – apurada como 149 bytes –, e das respostas que chegarem à EG, atribuídas como no caso anterior, demonstrada na Equação 9.

$$\begin{aligned}
 \text{TrafegoEG}_E &= S + (I * n) + \{[n/(n * 0,2)] * m\} + (R * n) \\
 &= 149 + (75 * n) + \{[n/(n * 0,2)] * 727\} + (3 * n) \\
 &= 78n + 3784
 \end{aligned}
 \tag{Equação 9}$$

Observando a análise teórica acima, pode-se afirmar que o tráfego produzido pelo protótipo SNMP ultrapassaria ao tráfego produzido pelo protótipo AM-E quando a quantidade de elementos gerenciados é maior que 946.

7.3. Análise do Tempo de Resposta

O tempo de resposta foi apurado em três condições diferentes de rede citadas na Seção 6.1.

Os tempos de resposta apresentaram muitas variações não somente com relação às condições de rede testadas, mas também com relação às abordagens de implementação.

7.3.1. Resultados na Gerência Local

O Quadro 7-4 mostra os tempos de resposta obtidos nos experimentos realizados na gerência local, situação onde não havia nenhum tipo de gargalo, ou seja, a latência da rede.

(apurada durante a realização dos experimentos) é menor que 1 milissegundo e a largura de banda mínima é de 10 Mbps.

Quadro 7-4: Tempos de resposta obtidos no experimento com Gerência Local, em milissegundos.

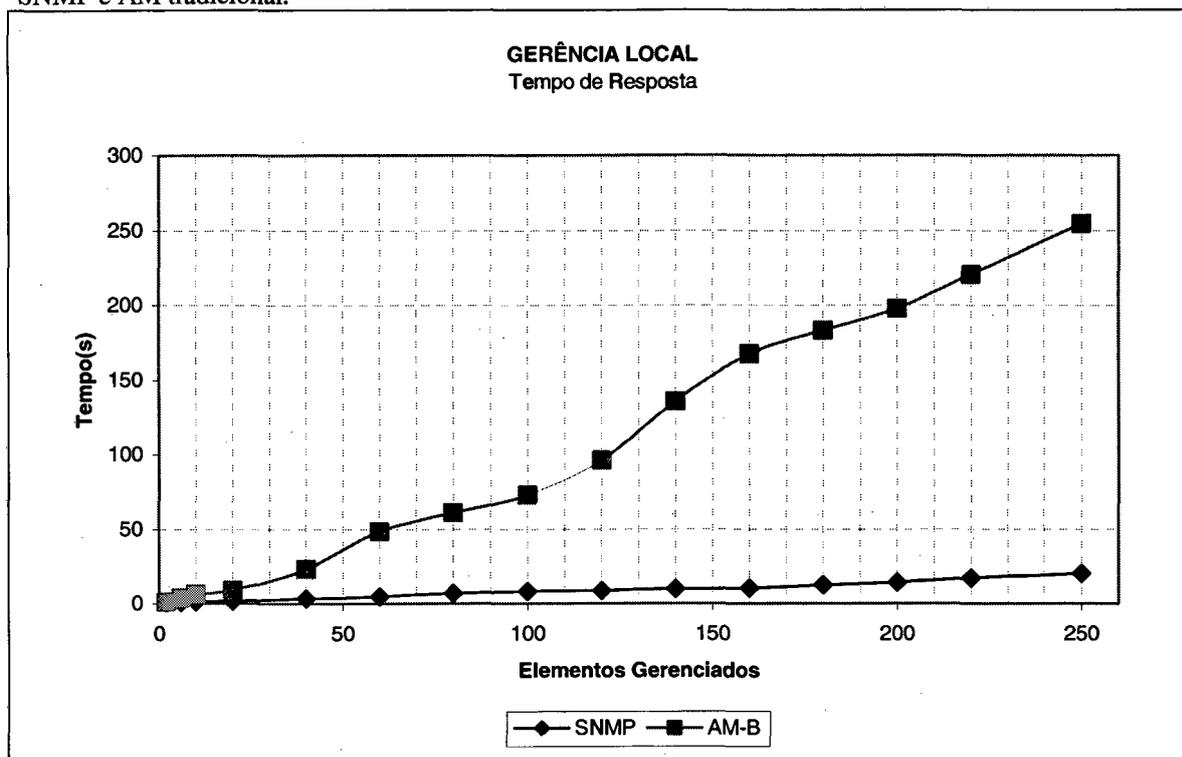
GERÊNCIA LOCAL				
Nós	SNMP	AM-B	AM-C	AM-D
2	280	1.376	1.146	797
6	980	3.775	2.786	2.180
10	1.096	6.427	4.681	4.520
20	1.768	9.117	8.450	11.358
40	3.276	23.094	16.957	13.537
60	4.924	48.461	43.129	18.547
80	6.674	60.893	50.903	30.567
100	8.223	72.807	74.646	46.565
120	8.623	95.925	80.436	50.192
140	9.913	149.801	135.965	51.358
160	10.256	167.525	154.181	76.540
180	12.591	183.177	175.057	103.747
200	14.129	198.022	191.197	131.758
220	17.235	220.485	210.012	142.667
250	20.227	254.702	248.350	177.854

Os tempos de resposta utilizados pelo protótipo SNMP são muito inferiores aos tempos conseguidos pelos protótipos de AM, devido à maior quantidade de bytes que necessita ser transferida para a execução dos AM e ainda, ao acréscimo do tempo de execução em cada um dos elementos gerenciados. A variação de tempo entre o SNMP e o AM tradicional, por exemplo, chega em algumas amostras a até 16 vezes.

Assim, em média, o protótipo SNMP concluiu a tarefa em somente 12,6% do tempo que o protótipo do AM tradicional utilizou para realizar a mesma tarefa, considerando o número de elementos gerenciados utilizados nos experimentos.

O gráfico que compara os valores referentes aos tempos obtidos com os protótipos SNMP e AM tradicional pode ser visto na Figura 7-7. Nele, é possível observar que a linha que representa o tempo de resposta do AM cresce mais rapidamente do que a linha que representa o tempo de resposta do SNMP, caracterizando que jamais os tempos de resposta do AM tradicional serão melhores, ou seja, inferiores aos tempos do protótipo SNMP nesta situação de rede específica.

Figura 7-7: Gráfico comparativo dos tempos de resposta obtidos no experimento da Gerência Local para o SNMP e AM tradicional.



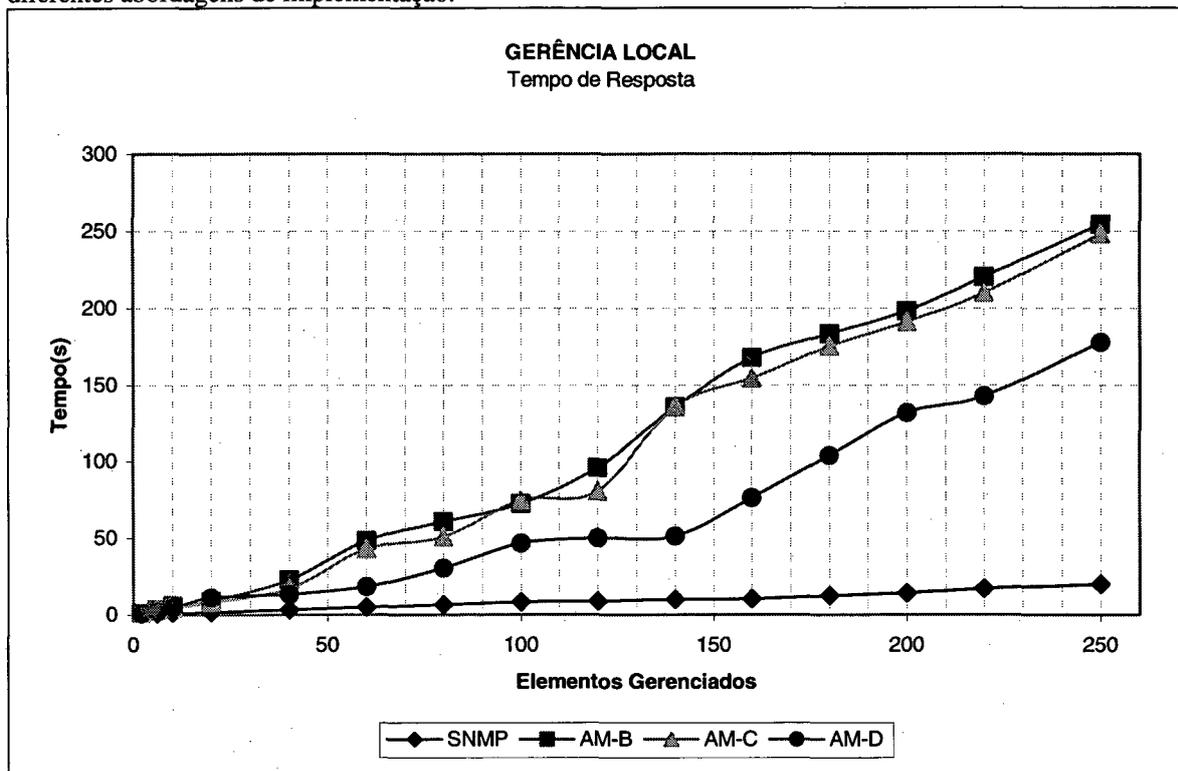
Com relação às outras abordagens de implementação, as mesmas conseguem reduzir os tempos de resposta conseguidos no protótipo tradicional do AM, mas também são muito superiores aos tempos obtidos com o protótipo SNMP, como demonstrado na Figura 7-8.

O protótipo AM-C conseguiu reduzir em média 11,82% o tempo de resposta em relação à implementação tradicional (AM-B), porém, seu tempo de resposta é ainda 9 vezes maior do que o tempo do protótipo SNMP. As maiores variações ocorreram nas amostras até 80 elementos gerenciados, nas quais algumas chegaram a ter uma diminuição de 27,70%.

Já o protótipo D, que descarrega parte de sua carga no decorrer da tarefa de gerenciamento, conseguiu tempos de resposta 39,22% e 30,56% menores do que os protótipos AM-B e AM-C respectivamente. Pode-se verificar também, que a maior diminuição aconteceu com um pequeno número de elementos gerenciados, onde ocorreram diferenças de até 57%.

Com relação aos tempos obtidos pelo SNMP, o protótipo AM-D apresenta tempos de resposta, em média, 5 vezes maiores.

Figura 7-8: Gráfico comparativo dos tempos de resposta obtidos no experimento da Gerência Local com diferentes abordagens de implementação.



7.3.2. Resultados do 1º Experimento na Gerência Remota

Apesar deste experimento (descrito na página 79) contar com equipamentos diferenciados na sua configuração, como *switches* e roteadores (equipamentos que contribuem para o aumento do tempo de latência), a latência apurada durante este experimento se manteve relativamente baixa (entre 2 e 10ms).

Apesar de ser maior do que a latência apurada no experimento com rede local, a latência não foi suficiente para que os protótipos de AM apresentassem melhores desempenhos, conforme os resultados mostrados no Quadro 7-5.

Quadro 7-5: Tempos de resposta obtidos no primeiro experimento com Gerência Remota, em milissegundos.

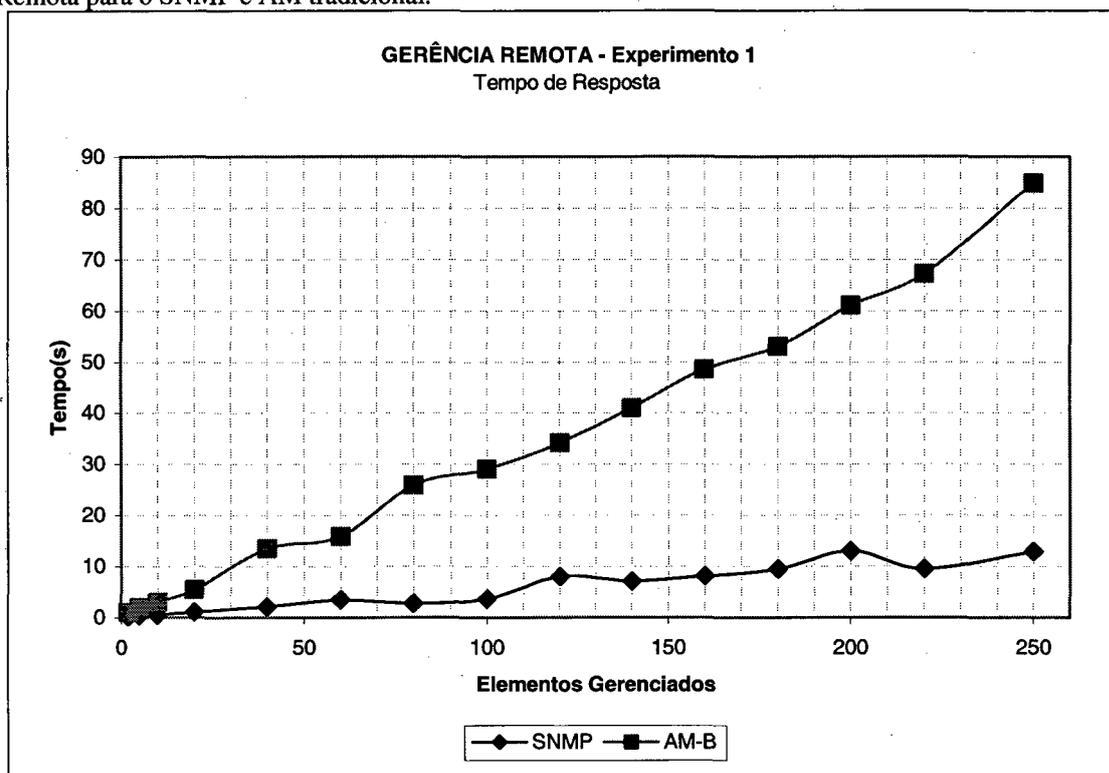
GERÊNCIA REMOTA – Experimento 1					
Nós	SNMP	AM-B	AM-C	AM-D	AM-E
2	175	1.013	426	404	341
5	341	2.008	1.151	1.094	921
10	557	3.003	2.237	2.125	1.790
20	1.107	5.528	4.544	4.317	3.635
40	2.098	13.463	9.196	8.736	7.357

GERÊNCIA REMOTA – Experimento 1					
Nós	SNMP	AM-B	AM-C	AM-D	AM-E
60	3.502	15.902	14.877	14.133	11.902
80	2.765	25.927	22.272	21.158	17.817
100	3.628	29.023	27.463	26.090	21.971
120	7.965	34.166	32.212	30.601	25.769
140	7.079	41.092	37.340	35.473	29.872
160	8.077	48.525	46.313	43.998	37.051
180	9.453	53.102	52.343	49.726	41.875
200	13.031	61.077	60.792	57.752	48.633
220	9.463	67.265	68.528	65.102	54.823
250	12.909	84.868	82.252	78.139	65.801

O protótipo SNMP consumiu, em média, somente 17,28% do tempo usado pelo protótipo do AM tradicional (AM-B) para realizar a mesma tarefa de recuperação de variáveis da MIB SNMP. Portanto, o tempo de resposta do AM tradicional foi, em média, 6 vezes maior do que o tempo gasto pelo protótipo SNMP.

Pode-se notar através da Figura 7-9 que o comportamento da curva de crescimento dos protótipos é semelhante ao obtido no experimento na gerência local, porque a configuração com menor banda passante afetou igualmente os dois protótipos, não provocando nenhuma característica diferenciada em nenhum deles.

Figura 7-9: Gráfico comparativo dos tempos de resposta obtidos no primeiro experimento da Gerência Remota para o SNMP e AM tradicional.



A diferença de tempos de resposta vista entre os gráficos da Figura 7-7 e da Figura 7-9 aconteceu devido ao fato que este segundo experimento foi realizado quando não havia nenhuma carga na rede, o que diminuiu muito os tempos de resposta apurados, pois o experimento em rede local sofreu mais os efeitos da carga nos elementos gerenciados.

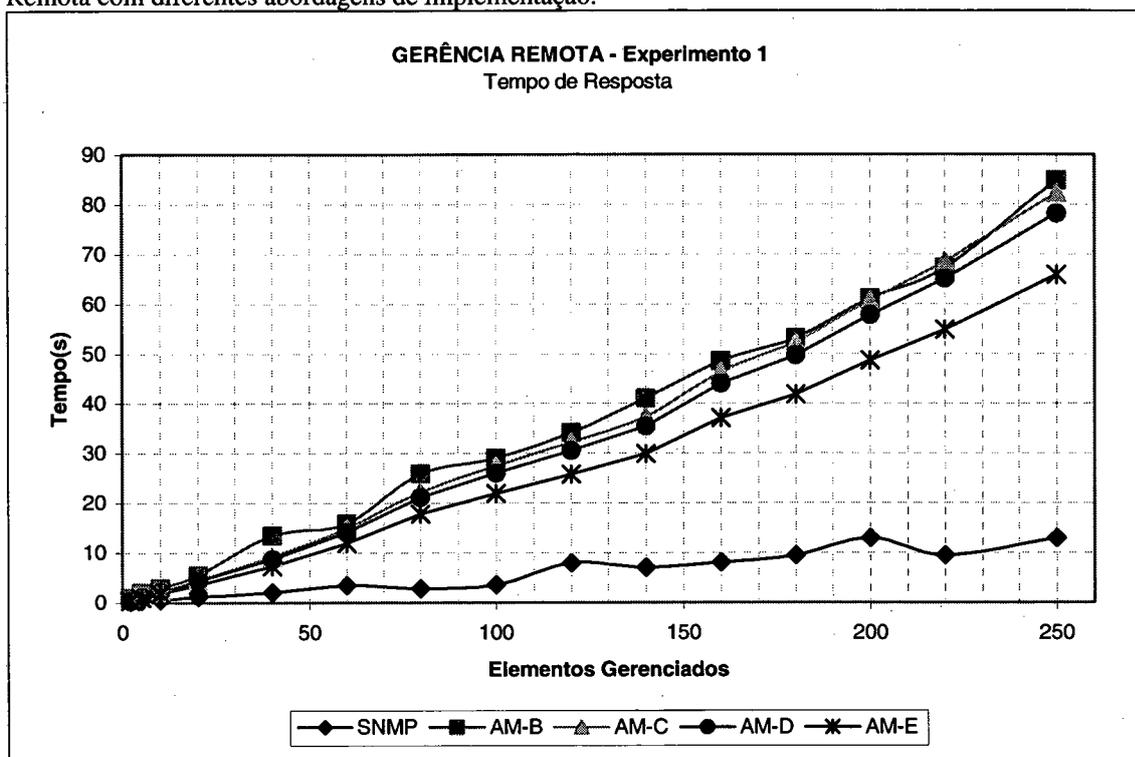
No primeiro experimento, o conjunto de equipamentos apresentava um tempo de resposta mais elevado devido ao volume de acessos a um dos equipamentos envolvidos, que realizava a função de servidor de impressão. Além disso, o experimento foi realizado em horário de pico de acessos à rede (horário comercial). Já o segundo experimento foi realizado em um período em um tráfego mínimo na rede (feriado), com equipamentos dedicados exclusivamente ao experimento e com configurações superiores aos utilizados no primeiro experimento.

Com relação às outras abordagens de implementação (AM-C, AM-D e AM-E), diferentemente do experimento anterior, onde apresentavam tempos de resposta melhores em relação ao AM tradicional (AM-B), os tempos de resposta neste experimento foram menos significativos, como pode ser visto na Figura 7-10.

Para o protótipo AM-C, até 40 elementos gerenciados o ganho de tempo de resposta em relação do AM tradicional (AM-B) foi, em média, de 35,13%. Acima de 40 elementos, a vantagem deste protótipo diminuiu para 4,84%.

Caso semelhante acontece com o protótipo AM-D, onde até 40 elementos, foram apurados tempos de resposta, em média, 38,38% menores do que o AM tradicional. Porém, acima de 40 elementos, sua vantagem gira apenas em torno de 9,60%.

Figura 7-10: Gráfico comparativo dos tempos de resposta obtidos no primeiro experimento com Gerência Remota com diferentes abordagens de implementação.



Já com o protótipo AM-E, sua diferença é mais significativa, com tempos de resposta, em média, 31,95% inferiores aos tempos obtidos com o AM tradicional. A variação dos tempos obtidos com o AM-E em relação ao protótipo AM tradicional se mostrou mais estável, onde a margem média de variação pode ser observada em praticamente todo o conjunto de amostragem.

7.3.3. Resultados do 2º Experimento na Gerência Remota

O segundo experimento (descrito na página 80) em uma situação de gerência remota foi realizado com valores médios de latência apurados entre 139,284 e 196,021 ms. Estes valores devem-se principalmente a conexão via modem que ligava à estação de

gerenciamento à rede onde estavam os elementos gerenciados. A banda passante desta conexão é de 28,8 Kbps, e havia um percentual de perda de pacotes de 5,3%.

Os valores obtidos dos experimentos estão listados no Quadro 7-6. Durante os experimentos pôde-se perceber a grande dificuldade do protótipo SNMP, que freqüentemente tinha que retransmitir pacotes que haviam sido perdidos ou que não eram respondidos dentro do período máximo permitido (*timeout*).

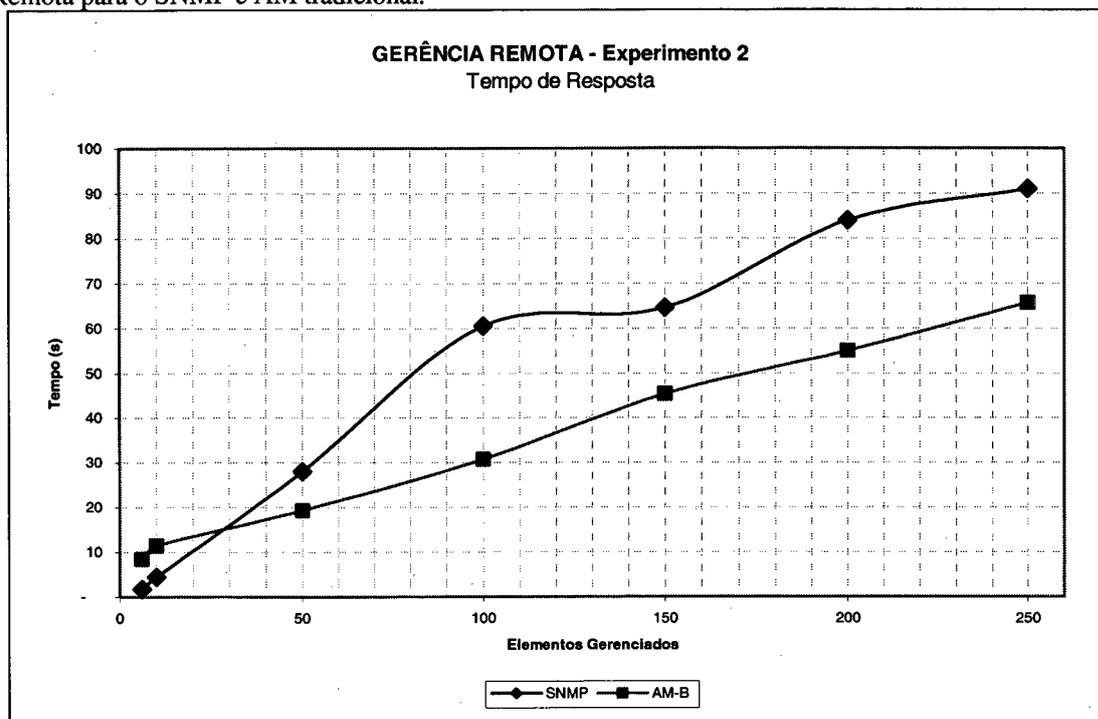
Quadro 7-6: Tempos de resposta obtidos no segundo experimento com Gerência Remota.

GERÊNCIA REMOTA – Experimento 2					
Nós	SNMP	AM-B	AM-C	AM-D	AM-E
6	1.726	8.512	5.335	4.180	2.840
10	4.425	11.505	5.373	6.443	6.877
50	28.059	19.275	13.073	11.013	10.010
100	60.475	30.705	25.574	18.415	17.826
150	64.480	45.420	34.108	26.708	21.592
200	83.960	54.953	43.185	36.601	31.014
250	90.990	65.666	54.320	42.864	37.501

Diante dos valores apresentados, o tempo de resposta do SNMP foi superior aos tempos obtidos com o protótipo AM tradicional (AM-B) já a partir de um número relativamente pequeno de estações, ou seja, menos que 50, com pode ser visto na Figura 7-11.

Acima de 50 elementos, os tempos de resposta obtidos pelo protótipo SNMP foram em média 34,49% maiores do que os tempos do protótipo do AM tradicional.

Figura 7-11: Gráfico comparativo dos tempos de resposta obtidos no segundo experimento da Gerência Remota para o SNMP e AM tradicional.



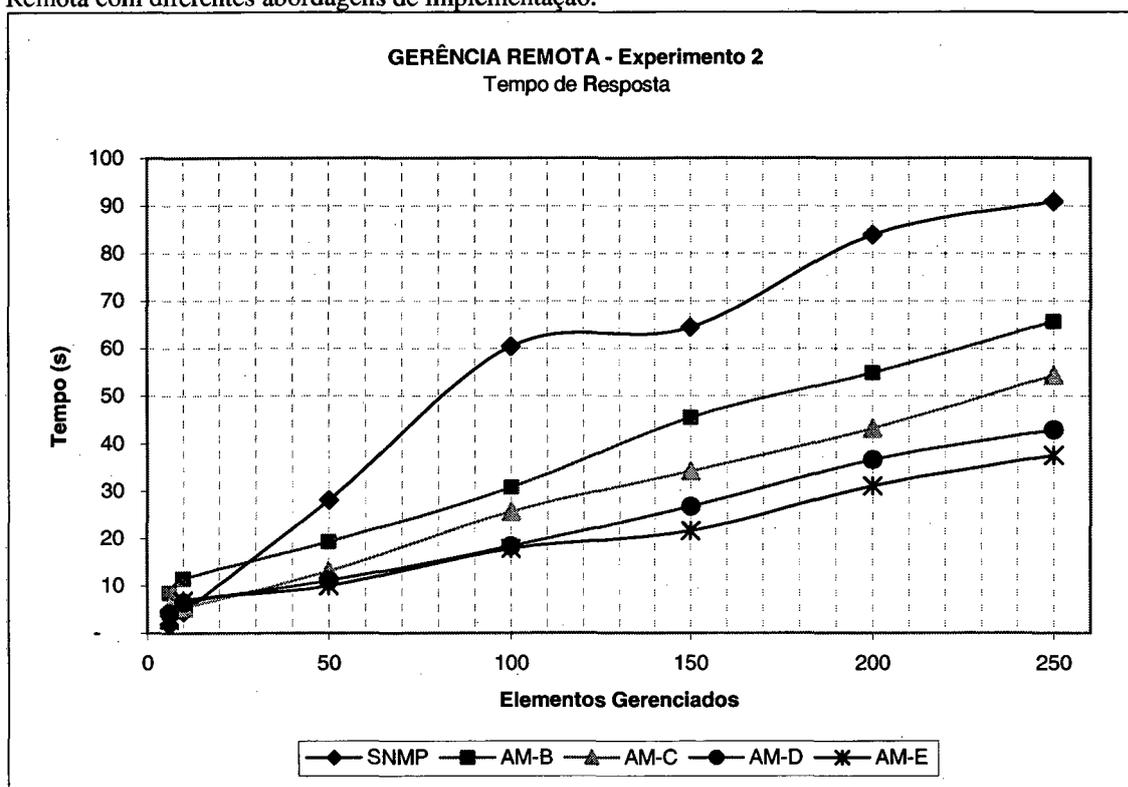
Estes altos tempos obtidos no protótipo SNMP que se diferenciam dos experimentos anteriores, devem-se ao fato de que toda a troca de mensagens necessária à obtenção das variáveis deve obrigatoriamente passar pelo segmento de gargalo, ao contrário dos AM, que passariam por este segmento somente uma vez.

Neste experimento, os tempos das abordagens AM-C, AM-D e AM-E também foram inferiores ao SNMP, e mostraram um desempenho melhor do que o AM tradicional, como pode ser visualizado na Figura 7-12.

Para o protótipo AM-C, por exemplo, o tempo médio de resposta acima de 50 elementos é quase a metade (49,92%) do tempo obtido com protótipo SNMP. Para os demais protótipos, os tempos também têm diferenças significativas favoráveis aos AM com 59,64% e 64,64% respectivamente para os protótipos AM-D e AM-E.

Nesta situação específica de rede, a adoção das diferentes abordagens de implementação (AM-C, AM-D e AM-E) se mostrou particularmente importante, conseguindo reduzir o tempo de resposta em relação à abordagem tradicional de AM (AM-B) em 29,02%, 41,01% e 47,97% respectivamente.

Figura 7-12: Gráfico comparativo dos tempos de resposta obtidos no segundo experimento com Gerência Remota com diferentes abordagens de implementação.



7.3.4. Comparações entre experimentos

Os valores de tempo de resposta com o experimento da rede local não puderam ser comparados diretamente com os valores obtidos nos demais experimentos devido aos seguintes fatores: ao efeito da carga na rede usada nos experimentos e ao uso de equipamentos com configurações diferentes. Estes fatores geraram valores de tempo de resposta bem mais altos do que dos experimentos com a rede remota, o que não permitiu sua comparação.

Comparando relativamente a variação que os tempos do protótipo SNMP tiveram em relação ao protótipo do AM tradicional nos dois experimentos, pode-se observar que:

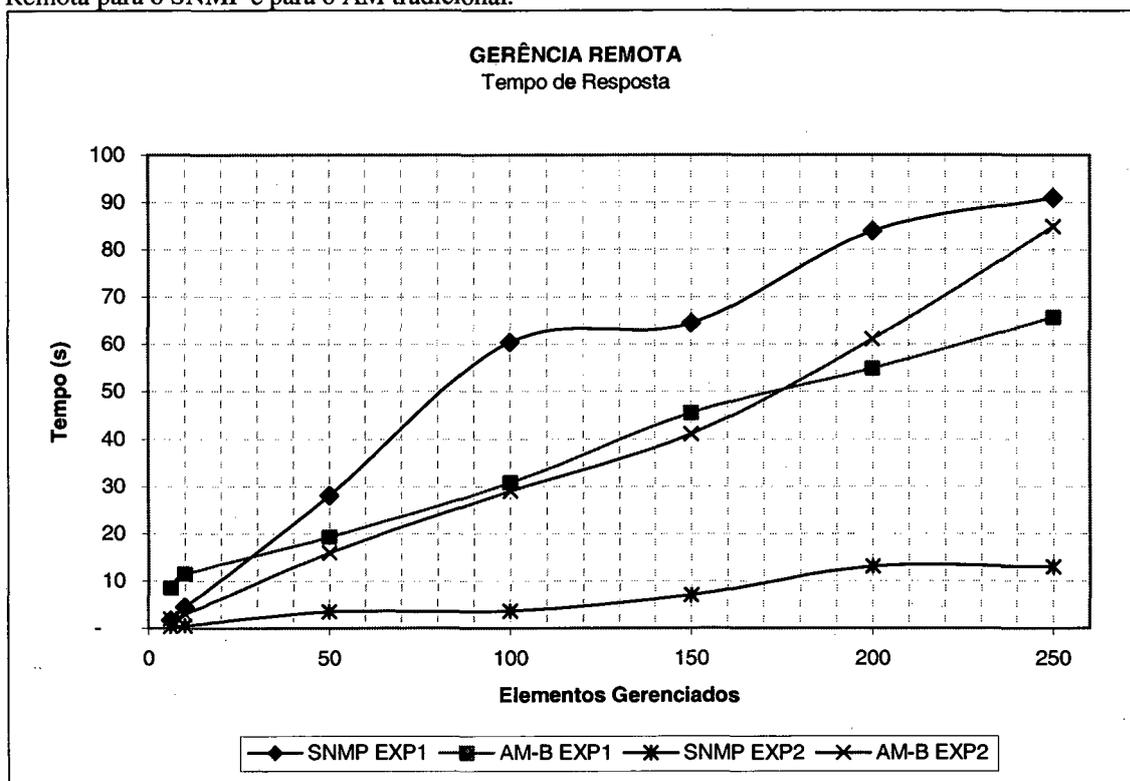
a) no primeiro experimento, onde a latência tende a zero, os tempos de resposta do protótipo do AM tradicional, foram em média 10,07 vezes maiores do que os tempos de resposta do protótipo SNMP;

b) no segundo experimento, onde a latência apurada foi ligeiramente maior (até 10 ms), a diferença de desempenho entre os dois protótipos foi bem menor, ou seja, o

protótipo de AM tradicional teve seus valores de tempo de resposta 6,03 vezes maior do que os tempos do protótipo SNMP, o que atesta uma redução desta margem de diferença, atribuída à variação da latência.

Quanto aos dois experimentos da Gerência Remota, pode-se verificar a diferença de comportamento entre os resultados, desta vez comparáveis, para os protótipos SNMP e AM tradicional, como é demonstrado na Figura 7-13.

Figura 7-13: Gráfico comparativo dos tempos de resposta obtidos no dois experimentos da Gerência Remota para o SNMP e para o AM tradicional.



O comportamento do protótipo do AM é semelhante nos dois experimentos, tendo apenas pequenas variações. Entretanto, o protótipo SNMP tem sua curva de comportamento totalmente diferente da apresentada no primeiro experimento devido à sua maior necessidade de transitar no enlace de gargalo, o qual no segundo experimento possui latência muito maior do que o primeiro.

7.4. Comparações com trabalhos anteriores

Os resultados obtidos neste trabalho confirmaram a apuração através de simulações realizada por Rubinstein (2001) e através de avaliação analítica realizada por Costa (1999), com relação:

a) ao efeito da latência, onde os resultados apurados apontam que o AM sofre menor influência da latência da rede do que o SNMP, e assim pode gerar melhores resultados nestas condições;

b) ao efeito da banda passante do enlace, onde os resultados apurados atestaram que os tempos de resposta crescem igualmente em todos os protótipos, o que reforça a afirmação de que as implementações sofrem igualmente influência da banda passante;

c) em redes locais e redes com gargalo com latência baixa, a diferença entre os tempos de resposta apurados nos experimentos com o SNMP e o AM tradicional foi idêntica a apurada aqui e o comportamento das curvas de crescimento foi semelhante.

Exclusivamente quanto às simulações executadas por Rubinstein (2001), o presente trabalho também confirmou que o uso de estratégias de descarregamento pode reduzir os tempos de respostas em relação à implementação tradicional e ao SNMP.

Entretanto, uma conclusão desenvolvida aqui diverge da conclusão obtida pelo estudo experimental executado no trabalho de Rubinstein (2001): os resultados dos experimentos que utilizam a implementação tradicional do AM, atestam naquele trabalho que a partir de um determinado número de elementos gerenciados (250), o número de *octetos* gerados pelo protótipo SNMP supera o tráfego gerado pelo protótipo do AM tradicional na estação de gerenciamento.

O experimento realizado aqui verificou que o AM tradicional sempre ocupará uma quantidade de recursos crescente, não permitindo que o tráfego na estação de gerenciamento gerado pelo protótipo SNMP seja maior em nenhuma circunstância.

Além do fato que os trabalhos possuem AM implementados com tamanhos iniciais muito diferentes, a provável causa para esta discordância foi a plataforma de implementação. O fator mais determinante neste sentido foi o vetor de itinerários do AM, que, implementado de acordo com a plataforma Aglets, cresce de acordo com a quantidade de elementos a serem visitados em uma razão de 75 bytes por elemento (valor muito alto em comparação ao tamanho das mensagens SNMP), o que não permitiu que o AM pudesse utilizar a mesma quantidade de recursos a medida em que a amostra de elementos do experimento aumentava.

8. CONCLUSÕES

A demora na coleta de algumas informações importantes pode causar problemas ou inviabilizar o diagnóstico correto do problema dentro da gerência de redes. Por outro lado, montar um sistema de gerenciamento que seja ágil e que permita ao administrador conhecer e acompanhar as condições de todos os dispositivos gerenciados sem consumir recursos preciosos é uma tarefa árdua.

Geralmente, a eficiência de um sistema de gerenciamento é conhecida quando todas as informações são obtidas e as ações necessárias são tomadas em tempo hábil e com o mínimo de interferência nas condições da rede. À medida que o sistema de gerenciamento não contempla a agilidade necessária, métodos alternativos devem ser adotados, tornando primordial uma análise destes casos visando encontrar alternativas viáveis e que tenham um custo-benefício compensador.

Cabe ressaltar, que as conclusões a seguir apresentadas advém especificamente dos protótipos desenvolvidos. A alteração da forma de implementação ou da plataforma adotada para o desenvolvimento dos AM pode acarretar mudanças significativas de comportamento nas variáveis de desempenho.

Conforme foi apurado nos experimentos, o uso de AM implementado de maneira tradicional, em tarefas de obtenção de variáveis da MIB causa uma maior utilização dos recursos da rede, com volumes de tráfego muito maiores em relação ao SNMP, seja para qualquer situação de rede. O uso de abordagens diferenciadas na implementação, que visem a economia de recursos é de extrema importância neste quesito (que consiste em um ponto negativo para a adoção do modelo), conseguindo minimizar este problema.

Para os protótipos desenvolvidos, somente pelo fato do AM não retornar à estação de gerenciamento é possível reduzir 16,3% do montante de tráfego apurado no AM tradicional. A adoção de uma estratégia de descarregamento pode reduzir em média 57% da utilização de recursos em comparação à abordagem tradicional, porém, o tráfego gerado pela utilização de um AM ainda permanece 8 vezes maior do que o tráfego gerado pelo SNMP.

Uma das motivações apregoadas como vantagens do uso de AM é a flexibilidade de implementação, que permite desenvolver soluções que possam poupar um segmento problemático ou a estação de gerenciamento e desta forma, conseguir melhores resultados.

Por exemplo, com relação ao tráfego trocado pela estação de gerenciamento em tarefas de obtenção de variáveis da MIB, os experimentos apontaram que o uso de um AM implementado de forma tradicional não consegue poupar a EG, fazendo com que a mesma precise trocar uma quantidade de bytes 75,71% maior do que o SNMP. Este efeito pode ser estendido a um segmento, que em uma situação hipotética de gerenciamento, necessite que a aplicação de gerenciamento faça circular o mínimo de informação por ele para que as aplicações normais da rede não sejam prejudicadas.

De acordo com os protótipos desenvolvidos, o uso de abordagens de implementação diferenciadas, como as descritas neste trabalho, pode evitar a sobrecarga da estação de gerenciamento ou do segmento problemático, gerando tráfego menor do que o SNMP a partir do número de elementos gerenciados identificados pelo estudo teórico. Mesmo na quantidade de elementos utilizada pelos experimentos, o uso destas abordagens reduziu a margem de diferença do tráfego em relação ao SNMP em até 41%.

Assim, o uso de abordagens de implementação não é somente opcional como necessário para que os AM consigam atingir seus objetivos de uma forma eficiente. É importante ressaltar que os experimentos e a análise com relação ao tráfego atestam que a abordagem tradicional de AM não consegue oferecer as vantagens postuladas, no sentido de evitar a sobrecarga da EG, na realização de tarefas de obtenção de variáveis, como a utilizada nos experimentos.

Com relação ao tempo de resposta em rede local, o uso de AM não se justifica pelos tempos de resposta apurados nos experimentos, em muito superiores aos tempos obtidos com o SNMP. Mesmo nesta situação de rede, o uso de abordagens alternativas pode reduzir em até 39,22% do tempo da abordagem tradicional, mas ainda se mantém superior, em média, 5 vezes aos tempos conseguidos pelo protótipo SNMP.

As diferenças apuradas entre os tempos de resposta dos protótipos – as quais dão larga vantagem ao SNMP – devem-se a dois fatores: ao excesso de tráfego produzido pelo AM em relação ao SNMP que precisa ser transferido entre os elementos; e ao tempo de

processamento da tarefa em cada um dos elementos gerenciados, que ajuda a incrementar os tempos de resposta.

O primeiro experimento com rede remota procurou simular uma situação em que a latência da conexão entre a estação de gerenciamento e os elementos gerenciados se mantém baixa, mas a largura de banda é reduzida. Porém, a diminuição da banda passante neste primeiro experimento não foi suficiente para imprimir alterações nos resultados dos protótipos, pois (exatamente como os trabalhos anteriores), todos os protótipos reagem de forma idêntica à diminuição da largura de banda. Mesmo assim, pôde-se notar que uma diminuição da diferença entre os tempos de resposta SNMP e do AM.

No segundo experimento em rede remota, foi observado o valor superior da latência no enlace de gargalo, além da banda passante também ser reduzida. Nesta situação, até mesmo o AM tradicional apresentou tempos de resposta menores do que o SNMP, na razão de 34,49%. O uso de diferentes abordagens conseguiu aumentar ainda mais esta vantagem.

Os experimentos confirmaram o que outros pesquisadores já haviam postulado: o que define o melhor desempenho do AM, e conseqüentemente, o pior desempenho do SNMP é o tempo de passagem pelo gargalo, mesmo que a rede local e os equipamentos envolvidos tenham alto desempenho.

Assim, para a gerência remota, o uso de AM se justifica para topologias onde exista significativa diferença de latência na conexão entre a estação de gerenciamento e os elementos. Mesmo em outros tipos de situações, o uso de AM pode ser justificado pela tentativa de se evitar a sobrecarga da estação de gerenciamento ou ainda, de um segmento que se deseja poupar.

Problemas complexos raramente são resolvidos por uma única técnica (Milojicic, 1999), por isso o uso de AM nas tarefas da gerência de redes adiciona vantagens táticas ao modelo SNMP, pois em determinadas situações – antes problemáticas – os AM podem ser adotados de forma a melhorar o tempo de resposta e permitir o uso mais racional de recursos da rede durante a execução de tarefas de gerenciamento. Os agentes móveis podem ser aplicados, inclusive para ampliar funcionalidades do sistema de gerenciamento.

As pesquisas na área e os experimentos realizados neste trabalho indicam ser viável a adoção de um modelo híbrido, que contemple a estrutura básica do modelo SNMP (com

agentes SNMP instalados em cada elemento gerenciado e a aplicação gerente sendo executada nas estações de gerenciamento), com AM circulando pela rede, executando somente as tarefas em situações nas quais eles realmente apresentem um custo-benefício vantajoso. As demais tarefas poderiam continuar sendo executadas de acordo com o modelo SNMP tradicional, que apresenta desempenho satisfatório em determinadas condições.

Os experimentos realizados neste trabalho contribuíram para comprovação de que a utilização de agentes móveis para tarefas de gerência de redes é vantajosa em alguns casos, porém, considera imprescindíveis melhorias no comportamento dos AM, com o objetivo de diminuir o tempo de resposta e principalmente a utilização de recursos da rede, de forma a conseguir melhor desempenho em determinadas condições em comparação ao SNMP.

Determinadas correntes de pesquisa (White, 1998) assinalam a grande probabilidade de que futuramente sejam desenvolvidos dispositivos de rede com capacidade de execução de aplicativos JAVA o que poderia ser um forte incentivo ao uso de soluções que utilizem AM para a gerência de redes.

Vale lembrar aqui que, o maior potencial de uso de AM na gerência de redes não se configura em tarefas de recuperação de variáveis, mas sim, em tarefas que podem ser executadas de forma autônoma. O desenvolvimento de aplicações para gerência SNMP que utilizem AM que podem adicionar novas facilidades ao gerenciamento de redes, ou permitir que algumas funcionalidades já existentes sejam executadas de maneira mais eficiente. Vários exemplos neste sentido – que inclusive podem ser aplicados na gerência de redes locais – foram mencionados na Seção 4.2.

8.1. Principais Contribuições

Resumidamente, este trabalho apresentou subsídios que validam as seguintes conclusões:

a) a gama de recursos disponíveis para o desenvolvimento de AM é grande e permite que sejam criadas aplicações que poderiam agregar muitas vantagens táticas e flexibilidade à gerência SNMP;

b) devido à flexibilidade de implementação, os AM podem ser implementados de maneiras diferentes visando melhorar requisitos necessários às funções da aplicação de gerência;

c) os AM podem apresentar melhor desempenho em relação à redução da sobrecarga na estação de gerenciamento somente se abordagens alternativas de implementação forem adotadas, ou seja, a abordagem de implementação de AM tradicional pode não oferecer esta vantagem. Para garantir este benefício, a escolha da plataforma de implementação é determinante;

d) o uso de abordagens de implementação de AM diferentes da tradicional não é tomado como uma opção e sim como um fator de sucesso para a adoção de AM em aplicações de gerência de redes, pois seus benefícios em relação à abordagem tradicional são evidentes depois dos experimentos realizados;

e) o uso de AM pode sobrecarregar a rede localmente, por isso, seu uso deve restringir-se a situações e tarefas específicas, onde o benefício é maior;

f) a largura de banda não é o principal fator que alavanca o melhor desempenho dos AM em relação ao SNMP e sim a latência;

g) em tarefas de recuperação de variáveis da MIB, as principais vantagens do uso de AM encontram-se em segmentos de rede separados da estação de gerenciamento por enlaces de gargalo com alta latência, tanto em relação à utilização da banda deste gargalo quanto em relação ao tempo de resposta.

8.2. Trabalhos Futuros

Como continuidade do trabalho, poderiam ser realizados novos experimentos com a intenção de apurar tempos de resposta em topologia *transit-stub*, que agrupa domínios de roteamento, ligados por um *backbone*.

Nesta situação de rede, encontrada nas redes *internet*, a estação de gerenciamento estaria localizada em um domínio *stub* e os elementos gerenciados, em outro domínio *stub*, onde ambos estariam ligados por um domínio *transit*. Esta situação pode ser localizada na RCT (Rede de Ciência e Tecnologia), que pode ser considerada um domínio *transit*, a qual

possui vários domínios *stubs* conectados a ela, como a própria FURB, a UNIDAVI (Universidade do Alto Vale do Itajaí) ou a UNIVALI (Universidade do Vale do Itajaí).

Outros experimentos poderiam ser realizados, no sentido de identificar qual a influência de cada fator de rede no desempenho dos AM em um número maior de situações de rede diferentes, o que não pode ser realizado por falta de diversidade de situações adequadas na rede utilizada nos experimentos.

Para conseguir melhores resultados com relação ao tráfego gerado, a forma de armazenamento do vetor de itinerários da plataforma *Aglets* poderia ser otimizada ou revista, de forma que cada endereço a ser visitado ocupe menos bytes.

Uma outra forma de extensão deste trabalho seria a realização de um estudo de viabilidade da utilização das propostas diferenciadas de implementação e dos recursos da arquitetura de AM vistos aqui em outros tipos de aplicações que atualmente utilizam-se da arquitetura cliente/servidor, inclusive em outras funções da gerência de redes.

Um direcionamento interessante seria a implementação de outras funcionalidades aos AM desenvolvidos, adicionando a eles mecanismos de detecção e prevenção de problemas mais comuns no gerenciamento de desempenho, implementando inclusive comportamento reativo baseado no conhecimento prévio, ou até um certo grau de inteligência.

O estudo do uso de AM dentro da gerência de redes também poderia ser realizado utilizando uma abordagem diferente: o uso de sistemas multiagentes com características de colaboração para monitoramento e controle, o que implicaria na descentralização de algumas das funções de gerência de redes.

9. REFERÊNCIAS BIBLIOGRÁFICAS

- (Aridor, 1998) ARIDOR, Y. e OSHIMA, M. Infrastructure of Mobile Agents: requirements and design. In *Proceedings of Mobile Agents: Second International Workshop MA'98* (set:1998). *Lecture Notes in Computer Science*, Vol. 1477. New York: Springer Verlag, 1998, p. 38-49. ISBN 3-540-64959-X.
- (Baldi, 1997) BALDI, M.; GAI, S. e PICCO, G.P. Exploiting Code Mobility in Decentralized and Flexible Network Management. In *Proceedings of Mobile Agents: First International Workshop MA'97* (abr:1997). *Lecture Notes in Computer Science*, vol. 1219, Berlim: Springer Verlag, 1997. p. 11-26. ISBN 3-540-62803-7.
- (Baumann, 1998) BAUMANN, J. e ROTHERMEL, K. The Shadow Approach: An orphan detection protocol for Mobile Agents. In *Proceedings of Mobile Agents: Second International Workshop MA'98* (set:1998). *Lecture Notes in Computer Science*, vol. 1477, New York: Springer Verlag, 1998. p. 2-13. ISBN 3-540-64959-X.
- (Baumann, 1999) BAUMANN, J. et al. MOLE – Concepts of a Mobile Agent System. In: MILOJICIC, D.; DOUGLIS, F. e WHEELER, R. Mobility: processes, computers and Agents. Massachusetts: Prentice-Hall, 1999. p. 536-554. ISBN 0-201-37928-7.
- (Bellavista, 2001) BELLAVISTA, P.; CORRADI, A. e STEFANELLI, C. Mobile Agent Middleware for Mobile Computing. *IEEE Computer Society Journal*, Los Alamitos (USA), vol. 34, nº 03, p. 73-81, mar 2001.
- (Bieszczad, 1999) BIESZCZAD, A. et al. Management of Heterogeneous Networks with Intelligent Agents. *Bell Labs Technical Journal*, Hartmut Peters, vol. 04, nº 04, p.109-135, out/dez 1999.
- (Bohoris, 2000) BOHORIS, C.; PAVLOU, G. e CRUICKSHANK, H. Using Mobile Agents for Network Performance Management. In *Proceedings of Seventh IEEE/IFIP Network Operations and Management Symposium NOMS 2000* (abr:2000). Honolulu: J.Hong e R. Weihmayer eds., 2000. p. 637-652. Doc. 16-3.
- (Braga, 2000) BRAGA, A.L. e XEXÉO, G. Agentes Móveis. Disponível em: <[http://www.cos.ufrj.br/~andre/Download/Agentes Moveis/Agentes Moveis - contemporanea.zip](http://www.cos.ufrj.br/~andre/Download/Agentes%20Moveis/Agentes%20Moveis%20-%20contemporanea.zip)>. Acesso em: jan 2000.

- (Braga, 2000a) BRAGA, A.L. Utilização de Agentes Móveis em Recuperação e Troca de Dados. Rio de Janeiro, 2000. Exame de qualificação para doutorado do Programa de Engenharia de Sistemas e Computação, Universidade Federal do Rio de Janeiro. Disponível em: <<http://www.cos.ufrj.br/~andre/Download/AgentesMoveis/eq.pdf>>. Acesso em: mai 2001.
- (Costa, 1999) COSTA, T.F.S. Avaliação Analítica do uso de Agentes Móveis para Gerência de Redes. Florianópolis, 1999. Dissertação de Mestrado do curso de Pós-Graduação em Ciências da Computação, Universidade Federal de Santa Catarina.
- (Fiorese, 2000) FIORESE, A. e WESTPHALL, C.B. Avaliação de Desempenho do Uso de Agentes Móveis na Gerência de Redes através de Técnicas de Medidas. In Anais do I Simpósio Catarinense de Computação (Ago:2000). Itajaí: Univali, 2000. p. 393-403.
- (Fünfroeken, 1998) FÜNFROCKEN, S. Transparent Migration of Java-Based Mobile Agents. In *Proceedings of Mobile Agents: Second International Workshop MA'98* (set:1998). *Lecture Notes in Computer Science*, vol. 1477. New York: Springer Verlag, 1998. p. 26-37. ISBN 3-540-64959-X.
- (Glitho, 1998) GLITHO, R. Management of Heterogeneous Networks. *IEEE Communications Magazine*, New York, vol. 36, nº 03, p. 34, mar 1998.
- (Goldszmidt, 1998) GOLDSZMIDT, G. e YEMINI, Y. Delegated Agents for Network Management. *IEEE Communications Magazine*, New York, vol. 36, nº 03, p. 66-70, mar 1998.
- (Gudwin, 2000) GUDWIN, R. Introdução à Teoria de Agentes. Material do curso homônimo ministrado no DCA – FEEC - UNICAMP. Disponível em: <<http://www.dca.fee.unicamp.br/~gudwin/courses/IA365G/>>. Acesso em: fev 2000.
- (Helden, 1999) HELDEN, J.A.M.V e MACHADO, M.W. Ferramenta de Gerenciamento de Falhas em roteadores utilizando WWW, Java e SNMP. Florianópolis, 1999. Trabalho de Conclusão do Curso de Bacharelado em Ciências da Computação, Universidade Federal de Santa Catarina.
- (Johansen, 1998) JOHANSEN, D. Mobile Agent Applicability. In *Proceedings of Mobile Agents: Second International Workshop MA'98* (set:1998). *Lecture Notes in Computer Science*, vol. 1477. New York: Springer Verlag,

1998. p. 80-98. ISBN 3-540-64959-X.
- (Lange, 1998) LANGE, D.B. Present and Future Trends of Mobile Agent Technology. In Proceedings of Mobile Agents: Second International Workshop MA'98 (set:1998). Lecture Notes in Computer Science, vol. 1477. New York: Springer Verlag, 1998. p. 01. ISBN 3-540-64959-X.
- (Lange, 1998a) LANGE, D.B. e OSHIMA, M. Programming and deploying Java Mobile Agents with aglets. Massachusetts: Addison-Wesley, 1998. 2ª ed. ISBN 0-201-32582-9.
- (Lange, 1999) LANGE, D.B. e OSHIMA, M. Mobile Agents with Java: The Aglet API. In: MILOJICIC, D.; DOUGLIS, F. e WHEELER, R. Mobility: processes, computers and Agents. Massachusetts: Prentice-Hall, 1999. p.495-512. ISBN 0-201-37928-7.
- (Mafinski, 1999) MAFINSKI, A. Protótipo de Software de Gerência SNMP para o ambiente windows NT. Blumenau, 1999. Trabalho de Conclusão do Curso de Bacharelado em Ciências da Computação. Universidade Regional de Blumenau.
- (Martin, 1994) MARTIN, J. e LEBEN, J. TCP-IP Networking: architecture, administration and programming. New Jersey: Prentice-Hall, 1994.
- (Mello, 2000) MELLO, J.L. Protótipo de um Agente SNMP para uma rede local utilizando a plataforma JDMK. Blumenau, 2000. Trabalho de Conclusão do Curso de Bacharelado em Ciências da Computação. Universidade Regional de Blumenau.
- (Milagres, 2001) MILAGRES, F.G. A aliança dos Agentes Móveis e Tecnologias contra os hackers. Developers' Magazine. Rio de Janeiro, vol. V, nº 54, p. 28-31, fev 2001.
- (Milojicic, 1998) MILOJICIC, D. et al. MASIF – The OMG Mobile Agent System Interoperability Facility. In Proceedings of Mobile Agents: Second International Workshop MA'98 (set:1998). Lecture Notes in Computer Science, vol. 1477. New York: Springer Verlag, 1998. p. 50-67. ISBN 3-540-64959-X.
- (Milojicic, 1999) MILOJICIC, D.; DOUGLIS, F. e WHEELER, R. Mobility: processes, computers and Agents. Massachusetts: Prentice-Hall, 1999. ISBN 0-201-37928-7.
- (Milojicic, 1999a) MILOJICIC, D. LAFORGE, W e CHAUHAN, D. Mobile Objects and Agents (MOA). In: MILOJICIC, D.; DOUGLIS, F. e WHEELER, R. Mobility: processes, computers and Agents. Massachusetts:

Prentice-Hall, 1999. p. 596-610. ISBN 0-201-37928-7.

- (Milojicic, 1999b) MILOJICIC, D. et al. MASIF – The OMG Mobile Agent System Interoperability Facility. In: MILOJICIC, D.; DOUGLIS, F. e WHEELER, R. *Mobility: processes, computers and Agents*. Massachusetts: Prentice-Hall, 1999. p. 629-641. ISBN 0-201-37928-7.
- (Oda, 1994) ODA, C. S. *Gerenciamento de Redes de Computadores: Noções Básicas*. Elaborado em set 1994. Disponível em: <<http://www.gter.cg.org.br/operacoes/gerencia-redes/doc0034a.em>>. Acesso em: fev 2000.
- (Pagurek, 2000) PAGUREK, B.; WANG, Y. e WHITE T. *Integration of Mobile Agents with SNMP: Why and How*. In *Proceedings of Seventh IEEE/IFIP Network Operations and Management Symposium NOMS 2000* (abr:2000). Honolulu: J.Hong e R. Weihmayer eds., 2000. Doc. 16-1.
- (Ricarte, 2000) RICARTE, I.L.M. *PooJava: Agentes*. Disponível em: <<http://www.da.fee.unicamp.br/courses/PooJava/agentes>>. Acesso em: dez 2000.
- (Rivalta, 2000) RIVALTA, P.C. *Mobile Agent Management*. Ottawa (Canadá), 2000. Dissertação de Mestrado - Faculty of Graduate Studies and Research - Department of Systems and Computer Engineering, Carleton University.
- (Rose, 1994) ROSE, M.T. *The simple book: an introduction to internet management*. 2ª ed. London: Prentice-Hall, 1994.
- (Rubinstein, 1999) RUBINSTEIN, M.G. e DUARTE, O.C.M.B. *Evaluating Tradeoffs of Mobile Agents in Network Management*. *Networking and Information Systems Journal*, Hermes Science Publications, vol. 2, nº 2, p. 237-252, fev 1999.
- (Rubinstein, 1999a) RUBINSTEIN, M.G.; DUARTE, O.C.M.B. *Reducing Response Time in Network Management Tasks by Using Mobile Agent Strategies*. Disponível em: <<http://www.gta.ufrj.br/producao/producao.html>>. Acesso em: dez 2000.
- (Rubinstein, 2000) RUBINSTEIN, M.G.; DUARTE, O.C.M.B. e PUJOLLE, G. *Using Mobile Agent Strategies for Reducing the Response Time in Network Management*. In *Proceedings of 16th IFIP World Computer Congress ICCT2000* (ago:2000). Beijing(China), 2000. Disponível em: <<http://www.gta.ufrj.br/producao/producao.html>>. Acesso em:

fev 2001.

- (Rubinstein, 2000a) RUBINSTEIN, M.G.; DUARTE, O.C.M.B. e PUJOLLE, G. *Reducing the Response Time in Network Management by Using Multiple Mobile Agents*. In *Managing QoS in Multimedia Networks and Services* (set:2000). Kluwer Academic Publishers, 2000. p. 253-265. ISBN 0-7923-7962-4.
- (Rubinstein, 2000b) RUBINSTEIN, M.G.; DUARTE, O.C.M.B. e PUJOLLE, G. *Evaluating the Network Performance Management Based on Mobile Agents*. In *Proceedings of Mobile Agents for Telecommunication Applications: Second International Workshop MATA'2000* (set:2000). *Lecture Notes in Computer Science*, vol. 1931. New York: Springer Verlag, 2000. p. 95-102. ISBN 3-540-41069-4.
- (Rubinstein, 2001) RUBINSTEIN, M.G. *Avaliação do Desempenho de Agentes Móveis no Gerenciamento de Redes*. Rio de Janeiro, 2001. Tese de doutorado do Programa de Engenharia Elétrica, Universidade Federal do Rio de Janeiro.
- (Sahai, 1998) SAHAI, A. e MORIN, C. *Enabling a Mobile Network Manager (MNM) through Mobile Agents*. In *Proceedings of Mobile Agents: Second International Workshop MA'98* (set:1998). *Lecture Notes in Computer Science*, vol. 1477. New York: Springer Verlag, 1998. p. 249-260. ISBN 3-540-64959-X.
- (Schill, 1998) SCHILL, A. et al. *An Agent Based Application for Personalized Vehicular Traffic Management*. In *Proceedings of Mobile Agents: Second International Workshop MA'98* (set:1998). *Lecture Notes in Computer Science*, vol. 1477. New York: Springer Verlag, 1998. p. 99-111. ISBN 3-540-64959-X.
- (Singhal, 1999) SINGHAL, S. e ZYDA, M. *Networked virtual environments: design and implementation*. Reading: Addison-Wesley, 1999. ISBN 0-201-32557-8.
- (Soares, 1995) SOARES, L.F.G.; LEMOS, G. e COLCHER, S. *Redes de Computadores: das LANs, MANs e WANs às redes ATM*. 2ª ed. Rio de Janeiro: Campus, 1995.
- (Spanhol, 2000) SPANHOL, F.A.; WATANABE, W.T. e RISO, B.G. *Aspectos de Operação e Gerenciamento de Redes Internet (TCP/IP)*. Anais do I Simpósio Catarinense de Computação (Ago:2000). Itajaí: Universidade do Vale do Itajaí - UNIVALI, 2000. p. 314-324.

- (Stallings, 1998) STALLINGS, W. SNMP and SNMPv2: The infrastructure for Network Management. *IEEE Communications Magazine*, New York, vol. 36, nº 03, p. 37-43, mar 1998.
- (Stallings, 1999) STALLINGS, W. *SNMP, SNMPv2, SNMPv3 and RMON 1 and 2*. 3ª ed. Reading MA: Addison-Wesley, 1999. ISBN 0-201-48534-6.
- (Starlin, 1998) STARLIN, G. e CARVALHO, A. *Guia Inteligente: Tecnologias de Redes – LAN's, WAN's, Protocolos, Serviços, Modelos e Internet*. Petrópolis: Book Express, 1998. ISBN 85-86846-05-8.
- (Sun, 2000) SUN Microsystems Inc. *Getting Started with the Java Dynamic Management Kit 4.2*. Part-number 806-6630. dez 2000. Disponível em: <<http://docs.sun.com>>. Acesso em: fev 2001.
- (Sun, 2000a) SUN Microsystems Inc. *Java Dynamic Management Kit 4.2. – Tools Reference*. Part-number 806-6631. dez 2000. Disponível em: <<http://docs.sun.com>>. Acesso em: fev 2001
- (Sun, 2000b) SUN Microsystems Inc. *Java Dynamic Management Kit 4.2. – Tutorial*. Part-number 806-6633. dez 2000. Disponível em: <<http://docs.sun.com>>. Acesso em: fev 2001.
- (Suri, 2000) SURI, N. et al. *Strong Mobility and Fine-Grained Resource Control in NOMADS*. In *Proceedings of the Agent Systems, Mobile Agents and Applications: Second International Symposium on Agent Systems and Applications and Fourth International Symposium on Mobile Agents ASA/MA 2000* (set:2000). *Lecture Notes in Computer Science*, vol. 1882. New York: Springer Verlag, 2000. p. 2-15. ISBN 3-540-41052-X.
- (Susilo, 1998) SUSILO, G.; BIESZCZAD, A. e PAGUREK, B. *Infrastructure for Advanced Network Management based on Mobile Code*. In *Proceedings of the IEEE/IFIP Network Operations and Management Symposium NOMS'98* (fev:1998). New Orleans: J.Hong e R. Weihmayer eds, 1998.
- (Tanenbaum, 1997) TANENBAUM, A.S. *Redes de Computadores*. Tradução da 3ª Ed. Rio de Janeiro: Campus, 1997. ISBN 85-352-0157-2.
- (Tomarchio, 2000) TOMARCHIO, O.; VITA, L. e PULIAFITO, A. *Nomadic User's Support in the MAP Agent Platform*. In *Proceedings of Mobile Agents for Telecommunication Applications: Second International Workshop MATA'2000* (set:2000). *Lecture Notes in Computer Science*, vol. 1931. New York: Springer Verlag, 2000. p. 233-241.

- (White, 1998) WHITE, T.; BIESZCZAD, A. e PAGUREK, B. *Distributed Fault Location in Networks Using Mobile Agents.* In *Proceedings of the 3th International Workshop on Agents in Telecommunications Applications IATA'98* (jul:1998). Paris, 1998.
- (White, 1999) WHITE, J.E. *Mobile Agents.* In: MILOJICIC, D.; DOUGLIS, F. e WHEELER, R. *Mobility: processes, computers and Agents.* Massachusetts: Prentice-Hall, 1999. p. 461-492. ISBN 0-201-37928-7.
- (Wittner, 2000) WITTNER, O.; HOELPER, C.J.E.; HELVIK, B.E. *Failure Semantics of Mobile Agent Systems involved in Network Fault Management.* In *Proceedings of Seventh IEEE/IFIP Network Operations and Management Symposium NOMS 2000* (abr:2000). Honolulu: J.Hong e R. Weihmayer eds, 2000. Doc. 16-1.
- (Wittner, 2000a) WITTNER, O.; HELVIK, B.E. *Simulating Mobile Agent Based Network Management using Network Simulator.* In *Proceeding of Second International Symposium on Agent Systems and Applications and Fourth International Symposium on Mobile Agents ASA/MA 2000* (set:2000). Zurich, 2000.